Mathematical Companion

for

Design and Analysis of Algorithms
(Unfinished!)

Jugal K. Kalita

Contents

Chapter 1

Algebra Basics

Basic and advanced algebra skills are play an important role in the analysis of algorithms. To analyze an algorithm, we must have a good understanding of how the algorithm functions. Once we understand the algorithm, we must be able to express its time or space needs in a mathematical manner. In doing so, algebra becomes important. In particular, concepts such as logarithms, partial fractions, factorials, proof by induction, and summation of series of various kinds become very useful. In addition, facility with asymptotic notation is fundamentally needed.

In this Chapter, we cover some of these topics. Other topics are covered in the rest of the book.

1.1 Proof by Induction

In analyzing algorithms, it is necessary to count the amount the time or space required by an algorithm as a function of the input size, and get a feel for how the amount varies with the input size, and see what happens when the input size becomes large. The input size is specified as a positive integer. For example, the input size for a sorting algorithm is the number of elements to sort. Many of the proofs when counting with integers can be done using proof by induction. Hence, it is important that we are comfortable in using induction for proofs.

Proof by induction follows from the *Axioms of Finite Induction* that is used to show that a set of positive integers contains all the positive integers if the following hold.

- The set contains 1, and
- If the set contains a positive integer k, then it also contains $k+1$.

When we deal with positive integers and need to prove a hypothesis, induction is a proof technique that we can often use. There are three main things we need to be concerned with: the *induction hypothesis*, the *base case* and the *induction step* or the *induction proof*.

Note that when we deal with proof by induction, it is possible that the induction hypothesis does not hold for a few of the smallest positive integers. For example, it is possible that the induction hypothesis holds for all positive integers $4, 5, 6, \cdots$, but not for the smallest three positive integers, *viz.*, $1, 2$ and 3. In such cases, we need to exclude the smallest positive integers from consideration. A specific example that can be provide by induction is that $n! > 2^n$ for all $n \geq 4$. That is, this result holds only for positive integers $4, 5, 6, \cdots$.

1.1.1 Sum of Cubes of Positive Integers

There are many interesting results with positive integers that can be proved by induction. One such result is that the sum of the cubes of the first n positive integers is equal to the square of the sum of the same first n positive integers. For instance,

$$1^3 + 2^3 + 3^3 + 4^3 + 5^3 = 1 + 8 + 27 + 64 + 125 = 225.$$

We also have that

$$(1 + 2 + 3 + 4 + 5)^2 = 15^2 = 225.$$

That is,

$$1^3 + 2^3 + 3^3 + 4^3 + 5^3 = (1 + 2 + 3 + 4 + 5)^2$$

This result is formalized in the following problem.

Problem 1 *Prove the following by induction.*

$$\Sigma_{i=1}^{n} i^3 = (\Sigma_{i=1}^{n} i)^2$$

Solution We need to identify the three basic elements of the proof.
Induction Hypothesis: It is what we need to prove, $\Sigma_{i=1}^{n} i^3 = (\Sigma_{i=1}^{n} i)^2$.
Base Case: We need to show that the induction hypothesis holds for the value of $n = 1$. This is easy to show. The left hand side of the equality is $\Sigma_{i=1}^{n}\ i^3$. When $n = 1$, the value of the left hand side is $\Sigma_{i=1}^{1}\ i^3 = 1^3 = 1$. The right hand side in this case is equal to $\left(\Sigma_{i=1}^{1} i\right)^2 = (1)^2 = 1$. Therefore, for the base case, the left hand side is equal in value to the right hand side. In other words, the induction hypothesis holds for the base case.
Induction Step: Here we need to show that if the induction hypothesis holds for an arbitrary positive integer value k (and for each positive integer smaller than k), it holds for the next higher integer $k + 1$. Let us assume that for some $k \geq 1$, the induction hypothesis holds. That is,

$$\Sigma_{i=1}^{k} i^3 = \left(\Sigma_{i=1}^{k} i\right)^2.$$

We need to show that

$$\Sigma_{i=1}^{k+1} i^3 = \left(\Sigma_{i=1}^{k+1} i\right)^2.$$

The proof follows. Let us start from the right hand side of the equation given immediately above.

$$\begin{aligned}
\left(\Sigma_{i=1}^{k+1} i\right)^2 &= \left[\Sigma_{i=1}^{k} + (k+1)\right]^2 \\
&= \left(\Sigma_{i=1}^{k} i\right)^2 + 2(k+1)\ \Sigma_{i=1}^{k} i + (k+1)^2 \\
&= \Sigma_{i=1}^{k}\ i^3 + 2(k+1)\frac{1}{2}k(k+1) + (k+1)^2 \\
&= \Sigma_{i=1}^{k}\ i^3 + (k+1)^2 k + (k+1)^2 \\
&= \Sigma_{i=1}^{k}\ i^3 + (k+1)^(k+1) \\
&= \Sigma_{i=1}^{k}\ i^3 + (k+1)^3 \\
&= \Sigma_{i=1}^{k+1}\ i^3
\end{aligned} \tag{1.1}$$

In the first step of the proof, we simply take the right hand side of what we need to prove it and rewrite it algebraically. That is, $\Sigma_{i=1}^{k+1} i^2$ is written as $\Sigma_{i=1}^{k} i^2 + (k+1)$ by separating out the last term. We then use the general formula for the expansion of $(a+b)^2$ to expand the square of an algebraic sum. In the third step of the proof, we use the assumption that the induction hypothesis holds for the integer k, *i.e.*, $\left(\Sigma_{i=1}^{k} i\right)^2 = \Sigma_{i=1}^{k} i^3$. In the third step we also use the fact that $\Sigma_{i=1}^{k} i = \frac{1}{2}k(k+1)$. Note that this is also a basic result that we assume we already know. The next several steps of the proof are algebraic simplifications.

1.1.2 Sum of Squares of Positive Integers

Sometimes, we need a formula to express the sum of the squares of the first n positive integers. The next problem gives us the necessary formula for this computation.

Problem 2 *Prove the following using induction.*

$$\Sigma_{i=1}^{n}\ i^2 = \frac{1}{6}n(n+1)(2n+1)$$

Solution Once again, to prove a hypothesis by induction, we need the three basic elements of the proof: the induction hypothesis, the base case, and the induction step.

Induction Hypothesis: We need to prove that $\Sigma_{i=1}^{n}\ i^2 = \frac{1}{6}n(n+1)(2n+1)$. This is called the induction hypothesis.

Base Case: We need to show that the induction hypothesis holds for the value $n = 1$. The left hand side of the hypothesis becomes $\Sigma_{i=1}^{1} i^2 = 1^2 = 1$. The right hand side of the hypothesis is $\frac{1}{6} \times 1 \times (1+1) \times (2+1) = \frac{1}{6} \times 2 \times 3 = 1$. Thus, the left hand side is equal to the right hand side. Therefore, the induction hypothesis holds for the base case.

Induction Step: We need to show that if the induction hypothesis holds for an arbitrary positive integer k (and each positive integer smaller than k), it then holds for the next higher integer $k+1$. Let us assume that induction hypothesis holds for k. This gives us the following.

$$\Sigma_{i=1}^{k} i^2 = \frac{1}{6}k(k+1)(2k+1)$$

We need to now show the following by increasing the upper limit of the sum from k to $k+1$.

$$\begin{aligned}\Sigma_{i=1}^{k+1} i^2 &= \frac{1}{6}(k+1)(k+2)[2(k+1)+1] \\ &= \frac{1}{6}(k+1)(k+2)(2k+3)\end{aligned}$$

The proof involves simple algebraic manipulation and follows.

$$\begin{aligned}\Sigma_{i=1}^{k+1} i^2 &= \Sigma_{i=1}^{k} i^2 + (k+1)^2 \\ &= \frac{1}{6}k(k+1)(2k+1) + (k+1)^2 \\ &= \frac{1}{6}(k+1)\left[k(2k+1) + 6(k+1)\right] \\ &= \frac{1}{6}(k+1)\left[2k^2 + k + 6k + 6\right] \\ &= \frac{1}{6}(k+1)\left[2k^2 + 7k + 6\right] \\ &= \frac{1}{6}(k+1)\left[2k^2 + 4k + 3k + 6\right] \\ &= \frac{1}{6}(k+1)\left[2k(k+2) + 3(k+2)\right] \\ &= \frac{1}{6}(k+1)(k+2)(2k+3) \qquad (1.2)\end{aligned}$$

This shows that the induction hypothesis holds for $k+1$.

The first step of the proof separates out the last term of the sum $\Sigma_{i=1}^{k+1}$. The second step uses the induction hypothesis to expand $\Sigma_{i=1}^{k} i^2$. The third step takes $\frac{1}{6}(k+1)$ as common between the two terms being added. The next several steps add up the terms within the square brackets and factorizes it to obtain $frac16(k+1)(k+2)(2k+3)$.

Using the result of the proof, we can obtains sums such as

$$\begin{aligned}1^2+2^2+3^2+4^2+5^2 &= \Sigma_{i=1}^{5} i^2 \\ &= \frac{1}{6}\times 5\times(5+1)\times(2\times 5+1) \\ &= \frac{1}{6}\times 5\times 6\times 11 \\ &= 55\end{aligned}$$

1.1.3 Divisibility of Sum of Cubes of Three Sequential Positive Integers

Problem 3 *Show that the sum of cubes of any three consecutive positive integers is divisible by* 9*. That is, for any integer* n*,* $n^3 + (n+1)^3 + (n+2)^3$ *is divisible by* 9*.*

Solution We need to write down the induction hypothesis, show that the induction hypothesis holds for the base case, and then show the induction step.
Induction Hypothesis: We need to show for any integer n, $n^3 + (n+1)^3 + (n+2)^3$ is divisible by 9. This is the induction hypothesis.
Base Case: When $n = 1$, $n^3 + (n+1)^3 + (n+2)^3 = 1^3 + (1+1)^3 + (1+2)^3 = 1 + 2^3 + 3^3 = 1 + 8 + 27 = 36$. Clearly, 36 is divisible by 9.
Induction Step: We assume that the induction hypothesis holds for a positive integer k (and for each positive integer smaller than k). Therefore, $k^3 + (k+1)^3 + (k+2)^3$ is divisible by 9. Now, we need to show that $(k+1)^3 + (k+2)^3 + (k+3)^3$ is divisible by 9 as well. The easiest way to show this is by showing that the difference of the two is divisible by 9.

$$\begin{aligned} & \left[(k+1)^3 + (k+2)^3 + (k+3)^3\right] - \left[k^3 + (k+1)^3 + (k+2)^3\right] \\ = \; & (k+3)^3 - k^3 \\ = \; & k^3 + 3 \times 3k(k+3) + 3^3 - k^3 \\ = \; & 9k(k+3) + 27 \\ = \; & 9\left[k(k+3) + 3\right] \\ = \; & 9\left(k^2 + 3k + 3\right) \end{aligned} \tag{1.3}$$

This clearly shows that the difference is divisible by 9. Since the original sum $k^3 + (k+1)^3 + (k+2)^3$ is divisible by 9 and the difference is divisible by 9 also, the sum $(k+1)^3 + (k+2)^3 + (k+3)^3$ is also divisible by 9. This completes our proof.
The result can be used to show that

$$100^3 + 101^3 + 102^3$$

is divisible by 9 without doing any of the calculations.

1.1.4 An Upper Bound on the nth Fibonacci Number

There is a positive integer sequence called the *Fibonacci sequence* that is sometimes useful in the analysis of algorithms. Let the ith Fibonacci number be denoted as F_i. By definition, we assume that $F_0 = 0$ and $F_1 = 1$ so that we know the first two numbers in the sequence. Once we know the first two Fibonacci integers, we obtain the rest of the sequence using the formula

$$F_n = F_{n-2} + F_{n-1} \tag{1.4}$$

Thus, the Fibonacci sequence evaluates to the integers

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \cdots$$

It is obviously an infinite series.

Problem 4 *If F_n is the nth Fibonacci number and $\phi = \left(\frac{1+\sqrt{5}}{2}\right)$, show that*

$$F_n \leq \phi^n$$

for each non-negative integer n.

Solution We will use induction for this proof.
Induction Hypothesis: $F_n \leq \phi^n$ where F_n is the nth Fibonacci number, $\phi = \left(\frac{1+\sqrt{5}}{2}\right)$, and n is a non-negative positive integer.
Base Cases: We need to show that $F_0 \leq \phi^0$ and $F_1 \leq \phi^1$ for the two smallest Fibonacci numbers. We have two base cases because the two smallest Fibonacci numbers are defined in a special manner, *i.e.*, without using a formula. This is easy. We know that $F_0 = 0$ by definition. We also know that $\phi^0 = \left(\frac{1+\sqrt{5}}{2}\right)^0 = 1$. Therefore,

$$F_0 \leq \phi^0.$$

Also, $F_1 = 1$ by definition. And, $\phi^1 = \left(\frac{1+\sqrt{5}}{2}\right)^1 = 1.616$. As a result,

$$F_1 \leq \phi^1.$$

Induction Step: We assume that the induction hypothesis holds for all non-negative integers equal to or smaller than a certain integer $k, k \geq 2$. In particular, $F_0 \leq \phi^0, F_1 \leq \phi^1, \cdots, F_k \leq \phi^k$. We need to show $F_{k+1} \leq \phi^{k+1}$ to prove our induction hypothesis.
In this case, it is necessary to use the formula that gives the kth Fibonacci number,

$$F_{k+1} = F_k + F_{k-1}.$$

We know that $F_k \leq \phi^k$ and $F_{k-1} \leq \phi^{k-1}$ by our assumption above. Therefore, we can write the following steps.

$$\begin{aligned} F_{k+1} &= F_k + F_{k-1} \\ &\leq \phi^k + \phi^{k-1} \\ &\leq \phi^{k-1}(\phi + 1) \\ &\leq \phi^{k-1}\left(\frac{1+\sqrt{5}}{2}\right) \\ &\leq \phi^{k-1}\left(\frac{3+\sqrt{5}}{2}\right) \\ &\leq \phi^{k-1}\phi^2 \\ &\leq \phi^{k+1} \end{aligned} \tag{1.5}$$

In the third step of theh proof above, we take ϕ^{k-1} as common and obtain $\phi^{k-1}(\phi+1)$ on the right hand side of the inequality. We have two factors ϕ^{k-1} and $\phi+1$. It is easy to show that $\phi^2 = 1+\phi$ as given below.

$$\begin{aligned}
\phi^2 &= \left(\frac{1+\sqrt{5}}{2}\right)^2 \\
&= \frac{1}{4}\left(1+5+2\sqrt{5}\right) \\
&= \frac{1}{4}\left(6+2\sqrt{5}\right) \\
&= \frac{1}{2}\left(3+\sqrt{5}\right) \\
&= 1+\frac{1}{2}\left(1+\sqrt{5}\right) \\
&= 1+\phi
\end{aligned}$$

This easily leads to the final step of the proof that $F_{k+1} \le \phi^{k+1}$.

Here, we note that mathematicians chose ϕ's value to be $\left(\frac{1+\sqrt{5}}{2}\right)$ because it is the root of the equation $\phi^2 = 1+\phi$ [Knu98, Page 13].

1.1.5 A Problem with Proof by Induction

A problem with proof by induction is that one needs to know what to prove before starting the proof. For example, to prove that

$$\Sigma_{i=1}^{n}\, i^2 = \frac{1}{6}n(n+1)(2n+1)$$

we need to know the right hand side of this equality to begin with. It may not be straightforward to find out what we exactly need to prove. In this specific case, if we actually want to find the formula $\frac{1}{6}n(n+1)(2n+1)$ for the sum, we need to use other ways to do so. Some such techniques are discussed in Section 2.3 of the book.

1.2 Partial Fractions

Sometimes, in solving problems in the analysis of algorithms, it is necessary to deal with fractions. Sometimes the fractions are complex and need to be reduced to simpler equivalent fractions before we can perform the computation needed. Often, the computation performed in analysis of algorithms is frequently the addition of a series. If the terms in the series that is being added are fractional, being able to obtain simpler fractions from complex ones may be useful when adding a more complex fraction. When a more complex fraction can be expressed as a sum (or difference) of simpler fractions, we call the simpler fractions using the term *partial fractions*.

Example 1 *Given $\frac{1}{k(k+1)}$, we want to be able to write it as the difference of two simpler fractions: $\frac{1}{k} - \frac{1}{k+1}$. Note that in the initial fraction, the denominator of the original fraction is a product of factors whereas in rewritten form, the two fractions are the simplest possible. In particular, in the rewritten form, the denominators of the fractions have no products.*

In this section, we look at a technique to obtain partial fractions from a complex fraction given to us. The method involves hypothesizing the partial fractions for a given complex fraction. In the hypothesized partial fractions, the numerators are not known, whereas the denominators are the simplest possible. We will explain the steps with an accompanying example problem.

Problem 5 *Obtain partial fractions for $\frac{1}{k(k+1)}$.*

Solution: The denominator of the fraction is the product of two factors: k and $k+1$. Therefore, we will hypothesize two partial fractions . We will make the assumption that the two partial fractions are added. The two partial fractions we hypothesize are $\frac{A}{k}$ and $\frac{B}{k+1}$. Thus, we assume the following.

$$\frac{1}{k(k+1)} = \frac{A}{k} + \frac{B}{k+1}$$

Here, A and B are constants whose values need to be determined. The values can be found without much difficulty in the following manner. Given the equality above, we can rewrite the right hand side and obtain.

$$\begin{aligned} \frac{1}{k(k+1)} &= \frac{A(k+1)+Bk}{k(k+1)} \\ 1 &= A(k+1)+Bk \end{aligned} \tag{1.6}$$

Equation 1.6 is obtained by multiplying both sides of the equation by $k(k+1)$. Given Equation 1.6, we can easily obtain the values of the constants A and B. One way to do so is by instantiating the equation for various values of k since the equation holds for all values of k. In particular, if $k = 0$, by instantiating Equation 1.6, we get

$$\begin{aligned} 1 &= A \times (0+1) + B \times 0 \\ 1 &= A \\ A &= 1 \end{aligned} \tag{1.7}$$

This gives us the value of A. We can set $k = -1$ in Equation 1.6 to obtain the value of B also.

$$\begin{aligned} 1 &= A \times (-1+1) + B \times (-1) \\ 1 &= A \times 0 - B \\ 1 &= -B \\ B &= -1 \end{aligned} \tag{1.8}$$

Now, that we know the values of A and B, we can write out the partial fractions completely as follows.

$$\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1} \tag{1.9}$$

Problem 6 *Obtain partial fractions for*

$$\frac{5x+6}{2-x-x^2}$$

Solution: The denominator is a function of x of degree 2. Therefore, the first step is to factor it into two linear factors.

$$\begin{aligned}\left(2-x-x^2\right) &= 2-2x+x-x^2\\ &= 2(1-x)+x(1-x)\\ &= (2+x)(1-x)\end{aligned}$$

Since the denominator can be written as the product of two linear factors in x, we can express the original fraction as a sum of two partial fractions each of which has a linear denominator and a constant numerator. Here, the two constants are A and B.

$$\begin{aligned}\frac{5x+6}{2-x-x^2} &= \frac{5x+6}{(2+x)(1-x)}\\ &= \frac{A}{2+x} + \frac{B}{1-x}\\ &= \frac{A(1-x)+B(2+x)}{(2+x)(1-x)}\\ 5x+6 &= A(1-x)+B(2+x) \qquad (1.10)\end{aligned}$$

Equation 1.10 is obtained by multiplying both sides of the equation by $(2+x)(1-x)$ or $2-x-x^2$. We need to now find the values of A and B. This we do just like the previous problem by instantiating the equation above for various values of x, in particular, $x=1$ and $x=-2$.

If $x=1$, the equation above gives the following.

$$\begin{aligned}5\times 1+6 &= A(1-1)+B(2+1)\\ 11 &= 3B\\ B &= \frac{11}{3} \qquad (1.11)\end{aligned}$$

Similarly, by assuming that $x=-2$ in the equation above, we get the value of A.

$$\begin{aligned}5\times(-2)+6 &= A(1+2)+B(2-2)\\ -4 &= 3A\\ A &= -\frac{4}{3} \qquad (1.12)\end{aligned}$$

Now, that we know the values of A and B, we have obtained our partial fractions.

$$\frac{5x+6}{2-x-x^2} = \frac{-4}{3(2+x)} + \frac{11}{3(1-x)}$$

Problem 7 *Sum the series:* $\Sigma_{k=1}^{k=n-1} \frac{1}{k(k+1)}$.

Solution: The first step is to find the partial fractions for $\frac{1}{k(k+1)}$. Then, when we instantiate the partial fractions for the various values of k, we find that most of the terms cancel out, giving us a nice tight expression for the sum.

$$\begin{aligned} \Sigma_{k=1}^{k=n-1} \frac{1}{k(k+1)} &= \Sigma_{k=1}^{k=n-1} \left[\frac{1}{k} - \frac{1}{k+1}\right] \\ &= \left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \cdots + \left(\frac{1}{n-1} - \frac{1}{n}\right) \\ &= 1 - \frac{1}{n} \end{aligned} \tag{1.13}$$

In obtaining this sum, we see that partial fractions are actually useful.

1.3 Logarithms

Logarithms play an important role in the study of algorithms. In analyzing algorithms for time and space needs, logarithms show up in expected as well as a unexpected places.

The *logarithm* of a real number with respect to a given real base, other than 1, is the index or exponent of the power to which the base must be raised to equal the number. Thus, if a, x and N are three real numbers such that

$$a^x = N$$

where $a > 0$ and $a \neq 1$, then x is called the *logarithm* of the number N with respect to the base a. We write this as

$$x = log_a\ N \tag{1.14}$$

For example, $9^2 = 81$, and therefore, $log_9\ 81 = 2$. Since, $3^4 = 81$, we can also write, $log_3\ 81 = 4$. Quite frequently, in algorithms analysis, we use base 2. For example, $2^{10} = 1024$ and therefore, we can write $log_2\ 1024 = 10$.

Frequently, we do not write the base explicitly when a relation is true for any base or the base in known to us.

1.3.1 Particular Cases

It is helpful to know the following particular cases:

- The logarithm of 1 to any finite non-zero base is zero. This is easy to show. Since

$$a^0 = 1$$

we have

$$log_a\ 1 = 0.$$

- Logarithm of a number to itself as base is 1. Since

$$a^1 = a,$$

we can write

$$log_a\ a = 1.$$

1.3.2 Rules of logarithm usage

To work with logarithms, a few basic rules of usage are useful.

1. *The logarithm of the product of two numbers is equal to the sum of the logarithms of the numbers.* In other words

$$log_a\ (m \times n) = log_a\ m + log_a\ n$$

This is easy to show. Let $x = log_a\ m$ and $y = log_a\ n$. Then, we can write $m = a^x$ and $n = a^y$. Therefore, we can write

$$m \times n = a^x \times a^y = a^{x+y}.$$

This gives us from the definition of logarithms,

$$log_a\ (m \times n) = log_a\ m + log_a\ n$$

2. *The logarithm of the quotient of two numbers is the difference of the logarithms of numerator and the denominator.* That is,

$$log_a\ \frac{m}{n} = log_a\ m - log_a\ n.$$

This is also easy to show. Let $x = log_a\ m$ and $y = log_a\ n$. Then, $m = a^x$ and $n = a^y$. Dividing, we get

$$\begin{aligned} \frac{m}{n} &= \frac{a^x}{a^y} = a^{x-y} \\ \Rightarrow log_a\ \frac{m}{n} &= log_a\ m - log_a\ n. \end{aligned} \tag{1.15}$$

3. *The logarithm of the power of a number is the product of the index of the number and the logarithm of the number.* In other words,

$$log_a\, m^n = n\; log_a\, m.$$

Once again, the proof is simple. Let $x = log_a\, m$ and $y = log_a\, m^n$. This gives us $m = a^x$ and $m^n = a^y$. We can further say that

$$\begin{aligned} y &= nx \\ \Rightarrow log_a\, m^n &= n\; log_a\, m. \end{aligned} \tag{1.16}$$

4. **Change of Base**: Sometimes it is necessary to change the base of a logarithm. This can be done using the formula given below.

$$log_a\, m = log_b\, m \times log_a\, n$$

Once again, this is easy to establish. Let $x = log_a\, m$, $y = log_b\, m$ and $z = log_a\, b$. This gives us $m = a^x$, $m = b^y$, and $b = a^z$. As a result,

$$a^x = m = b^y = (a^z)^y = a^{yz}$$

leading to

$$x = yz.$$

Thus, we can finally conclude

$$log_a\, m = log_b\, m \times log_a\, n$$

Logarithms show up in many places when we perform analysis of computer algorithms, simple or complex. For example, the well-known *binary search* algorithm that quickly finds a specific entry in a collection of data items takes logarithmic time. In particular, if the number of data items is N, the time taken to find a specific item or determine that it does not exist in the collection takes approximaely $log_2\, N$ time ([McC01, Chapter 2],[Lev03, Chapter 4],[BV00, Chapter 1],[CLRS01, Chapter 12],[Sed98, Chapter 5],[GT02, Chapter 3],[Knu98, pages 409-417]). Similarly, the analysis of sorting algorithms such as *merge sort* ([McC01, Chapter 3],[Lev03, Chapter 4],[BV00, Chapter 4],[CLRS01, Chapter 1],[Sed98, Chapter 8],[GT02, Chapter 4],[Knu98, pages 158-167]) and *quicksort* ([McC01, Chapter 3],[Lev03, Chapter 4],[BV00, Chapter 4],[CLRS01, Chapter 7],[Sed98, Chapter 7],[GT02, Chapter 4],[Knu98, Section 5.2]) shows that they take time proportional to $N\; log_2\, N$ where N is the number of data items to be sorted.

Prime numbers are of great interest in many computer algorithms, primarily ones that deal with cryptography or the ability to hide information from prying eyes, especially when the information is sent from one computer to another over the Internet. For example, such protection from unauthorized eavesdroppers is necessary in Internet commerce. There has been a lot of serious research on prime numbers over the past few centuries. There are many interesting facts about prime numbers. One such fact is that the number of prime numbers below a certain positive integer x has been shown to be approximately $\frac{x}{log\,x}$ ([GT02, Chapter 10], [MvV97, Chapters 3 and 4]). As a result, when many algorithms that work with prime numbers are analyzed for time needed, a logarithmic component shows up. For example, a traditional algorithm to find the prime numbers below a certain positive integer x called the Sieve of Eratosthenes can be shown to take time proportional to $x\ log\,log\,x$ time. A recent algorithm to find the primes below a certain integer x has been shown to take time which is less than (i.e., bound from above by) $x^{\frac{2}{3}}\ log\,x$ [LMO85].

Because logarithms appear so often in algorithms analysis, it is important that we are able to manipulate logarithms well.

1.4 Factorials

When working in analysis of algorithms, factorials show up often in our calculation. For a positive integer n, $n! = 1 \times 2 \times 3 \cdots (n-1) \times n$. The value of $n!$ increases very quickly as the value of n increases. So $3! = 1*2*3 = 6$, but $10! = 1*2*3*4*5*6*7*8*9*10 = 3,628,800$. Thus, an algorithm that takes exponential time or space with respect to the size of the input is a very inefficient algorithm. In practice, such algorithms cannot be used successfully unless the value of the input size is really small. Below are examples of a few algorithms that take exponential time with respect to the size of the input.

The idea of the factorial can be motivated by considering the permutations of a set of objects. For example, consider the alpha-numeric characters a, 1, and B. Suppose, we want to obtain all permutations possible. That is, we want to find all the ways in which the characters can be organized so that each character occurs only once.

There are three characters in total. The first position can be filled by any one of hte three characters leading to *three* possibilities. These are shown below. Note that the second and the third positions are unfilled at this time. An unfilled position is indicated by a dash (-).

$$\begin{array}{ccc} a & - & - \\ 1 & - & - \\ B & - & - \end{array}$$

Now, when we fill the second position, there are *two* possiblities for each case shown above. This is because the first position is already filled. When we fill the first position in three ways, and for each possibility in the first position, we fill the second position in two ways, we get $3 \times 2 = 6$ possibilities all of which are listed below. We have not decided what goes in the last position yet, and therefore, denote it by a dash.

$$\begin{array}{ccc} a & 1 & - \\ a & B & - \\ 1 & a & - \\ 1 & B & - \\ B & a & - \\ B & 1 & - \end{array}$$

We note that the last position can be filled in only way for each row above. This gives us a total of six possible permutations for the three characters a, 1, and B. They are all given below.

$$\begin{array}{ccc} a & 1 & B \\ a & B & 1 \\ 1 & a & B \\ 1 & B & a \\ B & a & 1 \\ B & 1 & a \end{array}$$

The product $3 \times 2 \times 1$ is usually written as $3!$ and is called the *factorial* of 3.

In general, if there are n objects to start with, and we want to obtain all possible permutations, we get $n!$ possibilities. This is because the first choice can be made in n ways; for each first choice, the second choice can be made in $n-1$ ways; for each first and second place choices, the third choice can be made in $n-2$ ways; etc. This gives us the total number of possible permutations as

$$n \times (n-1) \times (n-2) \cdots 3 \times 3 \times 2 \times 1$$

which is written as $n!$ in short.

1.5 Asymptotic Notation

We bother to analyze an algorithm only if we plan to use the algorithm often. If an algorithm is used only once, or on small amounts of data, it may not make sense to spend the time analyzing it. Thus, the usual assumption is that the amount of data the algorithm works on large. The amount of data usually means the number of inputs to an algorithm although it does not have to be so all the time.

When we analyze an algorithm, our goal is to produce a bound on the amount of time or space the algorithm takes when the size of the problem is n. As mentioned earlier, this generally means that the number of input is n.

Let us assume we are analyzing an algorithm for time requirement. The goal of this analysis is to find a bound on the amount of time the algorithm takes. There are three types of bounds we can find.

1. An upper bound which is usually written using a notation called the $O-$ notation .
2. A lower bound which is usually written using a notation called the $\Omega-$ notation.
3. A tight bound that is both an upper bound and a lower bound written using a notation called the $\Theta-$ notation.

We will look at each one of these bounds. Our primary objective is to find a tight or $\Theta-$ bound. However, sometimes, it is difficult to find a tight bound and in such a case, we are happy finding a lower or upper bound.

When we analyze an algorithm, the result of the analysis is a function in the number of inputs n. Thus, the various notations we mention above are actually functions.

Definitions...Incomplete

Once we have analyzed an algorithm, the result can be one of the following. Let us assume we are working with a $\Theta()$ type analysis.

- $\Theta(1)$: This means that the algorithm can be executed in constant time. That means the running time of the algorithm does not depend on the size on the input. An example of such an algorithm is calculating the interest payable I on a compound interest loan when we know the principal P borrowed, the rate of interest r in percentage and the period of the loan n in years. Here, the value of the interest payable is computed using the formula given below.

$$I = P\left(1 + \frac{r}{100}\right)^n$$

  Clearly, the value of n doesn't change the amount of time needed for this computation.

- $\Theta(log_2 n)$: This represents a sub-linear algorithm. An example of such an algorithm is the binary search algorithm. Note that a $\Theta(log_2 n)$ algorithm is a very slowly increasing function of n, almost constant. For example, if we have a billion inputs, the value of $log_2 n$ is about 20. Such an algorithm is way faster than a linear algorithm.

- $\Theta(\sqrt{n})$: Such an algorithm can be executed in $\Theta(\sqrt{n}$ time. This is also a sub-linear algorithm. An example of such an algorithm is Pollard's rho factoring algorithm for finding small factors of a composite integer [MvV97, page 91]. Pollard's rho factoring algorithm takes $\Theta(\sqrt{n}$ space as well. For an input of size a billion (10^9), the value of $\sqrt{n}$ is about 31623.

- $\Theta((log_2\ n)^4)$: There is an algorithm for finding the square root of an integer n (when we are using so-called modular arithmetic) in $\Theta((log_2; n)^4)$ time [MvV97, page 100]. When the two prime factors of n are known, an algorithm to find the square root can be executed in $\Theta((log_2; n)^3)$ time [MvV97, page 102].

There is a set of numbers called Carmichael numbers used in the discussion of prime numbers in public-key cryptography. A Carmichael number is defined as a composite integer such that $a^{n-1} \equiv 1(mod\ n)$ for all integers a that satisfy $gcd(a, n) = 1$. Without worrying about the definition, we just want to say here that researchers have showed that the number of Carmichael numbers $C(n)$ is given as

$$C(n) \leq n^{1-\{1+\Theta(1)\}\frac{ln\ ln\ ln\ n}{ln\ ln\ n}} \tag{1.17}$$

when $n \rightarrow \infty$ [MvV97, page 137]. . Here, we are not dealing with any analysis of algorithms, but this is being presented here as an example of a situation where researchers have come up with asymptotic expressions for availability of a specific kind of numbers.

There is an algorithm calle dhte Atkin's test or the Elliptic Curve Primality Proving Algorithm (ECPP) which has been shown to take approximately $\Theta\left((ln\ n)^6\right)$ time [MvV97, page 145].

- $\Theta(n)$: This means that the algorithm is linear. That is, it takes time proportional to the size of the input problem. An example of such an algorithm is finding the average of a set of n numeric inputs. To find the average, we need to sum the inputs and then divide the sum by the number n of inputs. To sum a set of numbers, we have to read the numbers one by one and add them together, then divide the total by the total number of numbers to be averaged, and the time taken is obviously proportional to the number of inputs. To find the average of 1000 numbers will take approximately 10 times required to find the average of 100 numbers.
- $\Theta(n\ log_2 n)$: An example of such an algorithm is sort algorithms such as quicksort or merge sort.
- $\Theta(n^2)$: An example of such an algorithm is a sorting algorithm such as bubble sort.
- $\Theta(n^3)$:If we have two $n \times n$ matrices and want to multiply them, it takes $\Theta(n^3)$ time.
- $\Theta(n^{2.87})$ or $\Theta(n^{log_2 n})$: There is a divide and conquer algorithm for matrix multiplication called the Strassen algorithm that takes $\Theta(n^{2.87})$ or $\Theta(n^{log_2 n})$ time [CLRS01, pages 735-742]. If we are trying to write a program that multiplies two matrices that are very large, say each of size $100,000 \times 100,000$, such an algorithm may be helpful although implementation issues become a consideration.
- $\Theta(n^3\ ln\ n)$: In Number Theory, there is a concept called *irreducible polynomials* over a field. [MvV97, page 156] discusses an algorithm to generate a random irreducible polynomial of a specific kind in $\Theta(n^3\ ln\ n)$ time.
- $\Theta(2^n)$ or $\Theta(n!)$: A lot of algorithms for real problems take exponential time or factorial time. It can be shown that asymptotically speaking, both are of the same order. There is a formula called the Stirling's approximation that relates $n!$ and powers of n.

There are several variations of this algorithm, but one that's commonly used is given below.

$$n! = \sqrt{2\pi\, n}\left(\frac{n}{e}\right)^n \tag{1.18}$$

1.6 Exercises

Prove the following using mathematical induction.

1. $2 + 4 + 6 + \cdots + 2n = n(n+1)$
2. $1^3 + 3^3 + 5^3 + \cdots + (2n-1)^3 = n^2\left(2n^2 - 1\right)$
3. $1 \times 2 + 2 \times 3 + 3 \times 4 + \cdots + n \times (n+1) = \frac{1}{3}n(n+1)(n+2)$
4. $1 \times 2 + 2 \times 2^2 + 3 \times 2^3 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
5. $1 \times 2 \times 3 + 2 \times 3 \times 4 + 3 \times 4 \times 5 + \cdots + n \times (n+1) \times (n+2) = \frac{1}{4}n(n+1)(n+2)(n+3)$
6. Show that $n! > 2^n$ for all $n \geq 4$.
7. Show that $9\left(9^n - 1\right) - 8n$ is divisible by 64.
8. Show that $5^{2n} + 3n - 1$ is divisible by 9.
9. Show that $7^{2n} - 48n - 1$ is divisible by 2304.
10. Show that for all $n \geq 7$, $n! \geq 3^n$.
11. Show that $a^n - b^n$ is divisible by $a - b$.
12. $\frac{1}{1\times 2} + \frac{1}{2\times 3} + \frac{1}{3\times 4} + \cdots + \frac{1}{n(n+1)} = \frac{n}{n+1}$
13. $\frac{1}{1\times 3} + \frac{1}{3\times 5} + \frac{1}{5\times 7} + \cdots + \frac{1}{(2n-1)(2n+1)} = \frac{n}{2n+1}$
14. $\frac{1}{1\times 4} + \frac{1}{4\times 7} + \frac{1}{7\times 10} + \cdots + \frac{1}{(3n-2)(3n+1)} = \frac{n}{3n+1}$

Chapter 2

Simple Summations

A loop is a very common programming structure used in a computer program. A loop allows a group of statements to be executed repeatedly until a certain termination condition is reached. There are several forms of loops in most programming languages. The most straight-forward loop consists of an integer index or counter that is incremented every time the loop statements are executed. These are usually called *for* loops. There are also other kinds of loops such as a *while* loop where the termination conditions can be arbitrary predicates. First, we look at the counter type loops.

A loop can be embedded inside another loop. There can be several levels of embedding in a complex computer program. To analyze programs or algorithms with loops, we need to obtain the sum of a series of numbers. Consider a very simple program given below that adds a sequence of integers.

Algorithm 1 *The following algorithm adds the positive integers from 1 to n.*

```
sum = 0;
for i=1 to n do
    sum = sum + i;
endfor
```

This program has a simple loop. The index of the loop or the counter is i. The value of i varies from 1 to n, incremented by one at a time. If we want to find the amount of time needed by such a program, we can express it as a sum of a series.

To determine computer time requirements for this loop, we need to identify a measure of the size of the problem the algorithm or the program is trying to solve. Here, the size of the problem can be considered to be the maximum value i can take, namely n. Assume the statements inside the loop take a constant amount of time c_2 to execute, and the statements outside take c_1 time. Also, assume that the amount of time taken by the algorithm for a problem of size n is indicated by $T(n)$. $T(n)$ can be expressed in the following manner.

$$T(n) = \sum_{i=1}^{n} c_2 + c_1 \tag{2.1}$$

Here, i is the index for the summation. THe term being added is c_2. c_2 does not depend on the index of summation i, and hence, itis a constant and is easily summable. We are simply adding c_2 over and over n times. Therefore, the sum is given as the following.

$$T(n) = nc_2 + c_1 \tag{2.2}$$

The function $T(n)$ is a polynomial in n of order 1. That is, it is linear. This is obvious since the loop goes only once over the numbers 1 through n. In analyzing algorithms, we are usually interested in the highest degree of the polynomial $T(n)$. Here, it is 1. Therefore, we can use the Θ-notation commonly used, to write the following.

$$T(n) = \Theta(n) \tag{2.3}$$

Here, the constant coefficient of c_2 of n is not written since it is not important beyond that the algorithm is linear. The constant c_1 is ignored as well.
Now, consider the algorithm given below.

Algorithm 2 *The following algorithm obtains the sum given below for an arbitrary n:*

$$1 \times 1 + 1 \times 2 + \cdots + 1 \times n + 2 \times 1 + 2 \times 2 + \cdots + 2 \times n + \cdots + n \times 1 + \cdots + n \times n$$

```
sum = 0
for i = 1 to n do
   for j = 1 to n do
       sum = sum + i * j
   endfor
endfor
```

This algorithm is easy to anlyze. Let $T(n)$ be the time taken by the algorithm for a problem of size n. Here, the size is measured in terms of the maximum value an index can take. The analysis is given below.

$$T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_2 + c_1 \tag{2.4}$$

Here, c_2 is a constant giving the amount of time taken by the algorithm statements inside the inner loop. c_1 is the time taken outside the two loops. The solution is given below. We sum over j first and then over i.

$$\begin{aligned} T(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n} c_2 + c_1 \\ &= \sum_{i=1}^{n} c_2 n + c_1 \\ &= c_2 \sum_{i=1}^{n} n + c_1 \\ &= c_2 n^2 + c_1 \\ &= \Theta(n^2) \end{aligned} \tag{2.5}$$

Here, c_2 is a constant and hence, can be taken outside the sum in the second step. Next, the index of summation is i, and the expression being added, n, does not depend on i. Therefore, the summation is straight-forward. Thus, $T(n)$ is a polynomial in n, the size of the problem at hand. The highest degree is 2. Therefore, it is a quadratic polynomial. If we are performing an asymptotic analysis, we can ignore all but the highest degree of n, and ignore all coefficients. The asymptotic analysis gives us the following $\Theta(n^2)$.

We all know how to multiply two matrices. Let us review the process, write an algorithm, and then analyze it for time complexity. Let us multiply two matrices $\mathbf{A}$ and $\mathbf{B}$, and obtain a result matrix $\mathbf{C}$. Let each of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ be $n \times n$ matrices. To see how we obtain $\mathbf{C}$, let us focus on a single element $\mathbf{C}_{ij}$ of $\mathbf{C}$, the element in the ith row and the jth column.

$$\begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ & & \vdots & & \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ & & \vdots & & \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\ & & \vdots & & \\ b_{i1} & \cdots & b_{ij} & \cdots & b_{in} \\ & & \vdots & & \\ b_{n1} & \cdots & b_{nj} & \cdots & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1n} \\ & & \vdots & & \\ c_{i1} & \cdots & c_{ij} & \cdots & c_{in} \\ & & \vdots & & \\ c_{n1} & \cdots & c_{nj} & \cdots & c_{nn} \end{bmatrix}$$

c_{ij} is obtained by taking the elements in the ith row of $\mathbf{A}$ and the jth column of $\mathbf{B}$, and multiplying them one pair at a time, and adding the products together. That is, we have the following.

$$\begin{aligned} c_{ij} &= a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} \\ &= \sum_{k=1}^{n} a_{ik}b_{kj} \end{aligned} \tag{2.6}$$

The computation of c_{ij} has to be done for all values of i from 1 to n, and for j from 1 to n. Thus, the algorithm for multiplying two matrices $\mathbf{A}$ and $\mathbf{B}$, and obtaining the matrix $\mathbf{C}$ can be written as follows.

Algorithm 3 *This algorithm multiplies two matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *to obtain a matrix* $\mathbf{C}$*.*

```
for i = 1 to n do
   for j = 1 to n do
       c[ij] = 0
       for k = 1 to n do
          c[ij] = c[ij] + a[ik] * b[kj]
       endfor
   endfor
endfor
```

Let $T(n)$ be the time taken to multiply two $n \times n$ matrices.

$$T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\sum_{k=1}^{n} c_2 + c_1 \right) \tag{2.7}$$

Here, c_2 is the time taken by the statement(s) inside the innermost loop. c_1 is the time required by statements inside the second loop, but not in the innermost loop. Obviously, after computing $T(n)$, we can determine that it is $\Theta * n^3)$.

2.1 Series

There are three different types of series that we are interested in the analysis of algorithms. They are the following.

- Arithmetic Series,
- Geometric Series, and
- Harmonic Series.

We will look at each of these three different series in this chapter.

2.2 Arithmetic Series

In an arithmetic series, there is a constant difference between two consecutive elements. The following is a general representation for an arithmetic series.

$$a, a+d, a+2d, a+3d, \cdots \tag{2.8}$$

The initial term is a. The difference between any two consecutive terms is d. If the series has n terms in it. The last term is $a + (n-1)d$. The value of the general term, or the ith term is $a + (i-1)d$, for an arbitrary $i, i \geq 1$.

Quite frequently, we need to obtain the sum of all the elements of such a series. Let the sum of an arithmetic series with n elements be S_n. Therefore, we can write the following.

$$S_n = \sum_{i=1}^{n}(a + (i-1)d) \tag{2.9}$$

To obtain this sum, we can proceed in the following manner. We write the sum in the expanded format.

$$S_n = a + (a+d) + (a+2d) + \cdots + (a+(n-2)d) + (a+(n-1)d) \tag{2.10}$$

We write the same sum again, but this time, we write the elements in reverse order.

$$S_n = (a+(n-1)d) + (a+(n-2)d) + \cdots + (a+2d) + (a+d) + a \tag{2.11}$$

We add the two equations term by term, simplify and solve for S_n. The process is illustrated below.

$$\begin{array}{rcccccccccc} S_n & = & a & + & (a+d) & + & \cdots & + & (a+(n-2)d) & + & (a+(n-1)d) \\ S_n & = & (a+(n-1)d) & + & (a+(n-2)d) & + & \cdots & + & (a+d) & + & a \\ \hline 2S_n & = & (2a+(n-1)d) & + & (2a+(n-1)d) & + & \cdots & + & (2a+(n-1)d) & + & (2a+(n-1)d) \end{array}$$

The term on the right hand side, namely, $2a + (n-1)d$, occurs n times. Therefore, we can write the following.

$$\begin{aligned} 2S_n &= n(2a + (n-1)d) \\ S_n &= \frac{1}{2}n(2a + (n-1)d) \end{aligned} \tag{2.12}$$

This is the general formula that gives us the sum of the first n elements of an arithmetic series.

Problem 8 *Sum the first n positive integers.*

Quite frequently, we need to obtain the sum of the first n positive integers, for an arbitrary value of n. That is, we need a compact formula for S_n given below.

$$S_n = 1 + 2 + \cdots + (n-1) + n \tag{2.13}$$

or,

$$S_n = \sum_{i=1}^{n} i \tag{2.14}$$

This is clearly an arithmetic series whose first element is 1. The difference between two consecutive elements is 1 as well. There are n terms in the series. Therefore, we can go ahead and use the general formula for the sum of n elements of an arithmetic series. The particulars are given below.

$$\begin{aligned} a &= 1 \\ d &= 1 \end{aligned}$$

Using the general formula for the sum of an arithmetic series, the sum S_n is given as

$$\begin{aligned} S_n &= \frac{1}{2}n(2 \times 1 + (n-1) \times 1) \\ &= \frac{1}{2}(2 + n - 1) \\ &= \frac{1}{2}n(n+1) \end{aligned} \tag{2.15}$$

This sum can be visualized in another way.

$$S_n = 1 + 2 + \cdots + (n-1) + n \tag{2.16}$$

There are n elements in the series. The first and the last elements are 1 and n, respectively. They add up to $n+1$. Thus, the average value of these two elements is $\frac{n+1}{2}$. The second and the last but one elements are 2 and $(n-1)$, respectively. They add up to $2 + (n-1) = n+1$ again. The average of these two elements is $\frac{n+1}{2}$ again. Similarly, we can show that the average value of any two elements at equal distance from the front and the end is $\frac{n+1}{2}$. In case there are an odd number of elements, the middle element stands out by itself and its value is $\frac{n+1}{2}$. Thus, we have n elements with the average value $\frac{n+1}{2}$. Therefore, the sum of the series is

$$\begin{aligned} S_n &= n \times \text{the value of an average element} \\ S_n &= \frac{1}{2}n(n+1) \end{aligned} \tag{2.17}$$

The above observation can be written in a slightly more formal manner. The ith element of the series, counted from the beginning is i. The corresponding "last but ith" element of the series, counted from the back is $(n - (i-1))$. Therefore,

$$\begin{aligned} S_n &= n \times \text{the value of an average element} \\ &= n \times \frac{1}{2}(i + (n - (i-1))) \\ &= \frac{1}{2} \times (i + n - i + 1) \\ &= \frac{1}{2}n(n+1) \end{aligned} \tag{2.18}$$

Algorithm 4 *The following algorithm obtains the sum of the following series:*

$$1 \times 1 + 2 \times 1 + 2 \times 2 + 3 \times 1 + 3 \times 2 + 3 \times 3 + \cdots + n \times n$$

```
sum = 0
for i = 1 to n do
   for j = 1 to i do
       sum = sum + i * j
   endfor
endfor
```

Let us say that the size of this algorithm is given in terms of n, the highest value of a loop index. The time taken is given as $T(n)$. We can write $T(n)$ as given below. Let c_2 be the time taken by the instruction(s) inside the inner loop. Let c_1 be the time taken by the instruction(s) outside the loop. Here, the index of the first sum is i, and the index for the second sum is j. We start summing from inside.

$$\begin{aligned} T(n) &= \sum_{i=1}^{n}\sum_{j=1}^{i} c_2 + c_1 \\ &= \sum_{i=1}^{n} c_2 i + c_1 \\ &= c_2 \sum_{i=1}^{n} i + c_1 \\ &= \frac{1}{2}c_2 n(n+1) + c_1 \\ &= \frac{1}{2}c_2 n^2 + \frac{1}{2}c_2 n + c_1 \\ &= \Theta(n^2) \end{aligned} \tag{2.19}$$

c_2 does not depend on the index i. Thus, the inner summation is removed easily. c_2 is a constant that does not depend on the index i, and therefore can be taken outside the summation. Next, we need to sum the arithmetic series $\sum_{i=1}^{n}$. We know the sum to be $\frac{1}{2}n(n+1)$. Finally, we write it out as a polynomial in n. It is a quadratic polynomial. In terms of asymptotic notation, the analysis can be written as $\Theta(n^2)$.

2.3 Summing up the Series: $\sum_{i=1}^{n} i^k$

Very often, when analyzing an algorithm with emebedded loops, we have to add a series of the type $\sum_{i=1}^{n} i^k$ where k is a small positive integer. This sum occurs when we have $k+1$ loops, one inside the other. Normally, we do not have efficient algorithms if they take more than n^3 time.

Below, we sum the series $\sum_{i=1}^{n} i$ once again to show a technique that can be used to obtain such sums. We later extend the same technique to computing the value of $\sum_{i=1}^{n} i^2$. The technique can be used to compute $\sum_{i=1}^{n} i^3$, etc., but the algebraic computation becomes too clumsy.

2.3.1 Computing $\sum_{i=1}^{n} i$ another way

To compute $\sum_{i=1}^{n} i$, we can proceed by computing the value of the expression $(k+1)^2 - k^2$, and using it repeatedly. First, we compute $(k+1)^2 - k^2$.

$$\begin{aligned} (k+1)^2 - k^2 &= k^2 + 2k + 1 - k^2 \\ &= 2k + 1 \end{aligned} \tag{2.20}$$

We will instantiate this equation for the value of k going down to 1 from n. For example, when the values of k are 1 and 2, we get the following, respectively.

$$\begin{array}{ccccccc} 2^2 & - & 1^2 & = & 2\times 1 & + & 1 \\ 3^2 & - & 2^2 & = & 2\times 2 & + & 1 \end{array}$$

We instantiate up to a value n for k, and add all the equations together. This step is shown below.

$$\begin{array}{llllll} 2^2 & - & 1^2 & = & 2\times 1 & +1 \\ 3^2 & - & 2^2 & = & 2\times 2 & +1 \\ 4^2 & - & 3^2 & = & 2\times 3 & +1 \\ & & & \vdots & & \\ n^2 & - & (n-1)^2 & = & 2\times(n-1) & +1 \\ (n+1)^2 & - & n^2 & = & 2\times n & +1 \\ \hline (n+1)^2 & - & 1^2 & = & 2\times(1+2+3+\cdots+n) & +n \end{array}$$

Note that the intermediate terms on the left side cancel out. Now, the expression $1 + 2 + 3 + \cdots + n$ is the one whose value we want to obtain. Let us call this sum S_n and simplify to obtain its value.

$$\begin{aligned} (n+1)^2 - 1 &= 2S_n + n \\ \Rightarrow 2S_n &= (n+1)^2 - n - 1 \\ &= n^2 + 2n + 1 - n - 1 \\ &= n^2 + n \\ &= n(n+1) \\ \Rightarrow S_n &= \frac{1}{2}n(n+1) \end{aligned} \tag{2.21}$$

Note that the results is the same expression we got as when the value was obtained by considering it to be an arithmetic series. This approach can be used to obtain the value of $\sum_{i=1}^{n} i^2$. Here, to start with, we need to obtain an expression that gives the difference between the cubic powers of two consecutive integers. Let the integers be $k+1$ and k. We need to remember that, in general, the following holds.

$$(a+b)^3 = a^3 + 3ab(a+b) + b^3 \tag{2.22}$$

The difference we need is computed next.

$$\begin{aligned}(k+1)^3 - k &= k^3 + 3k(k+1) + 1 - k^3 \\ &= k^3 + 3k^2 + 3k + 1 - k^3 \\ &= 3k^2 + 3k + 1 \end{aligned} \tag{2.23}$$

We will instantiate this equation for values of k ranging from 1 to $n+1$. We will then add them all up and rearrange them algebraically to obtain an expression for the sum we are looking for.

$$\begin{array}{lclclcl} 2^3 & - & 1^3 & = & 3 \times 1^2 & + & 3 \times 1 & +1 \\ 3^3 & - & 2^3 & = & 3 \times 2^2 & + & 3 \times 2 & +1 \\ & & & \vdots & & & & \\ n^3 & - & (n-1)^3 & = & 3 \times (n-1)^2 & + & 3 \times (n-1) & +1 \\ (n+1)^3 & - & 1 & = & 3 \times n^2 & + & 3 \times n & +1 \\ \hline (n+1)^3 & - & 1 & = & 3 \times (1^2 + 2^2 + \cdots + n^2) & + & 3 \times (1 + 2 + \cdots + n) & +n \end{array}$$

We know the sum $1 + 2 + \cdots n = \frac{1}{2}n(n+1)$. We want to obtain the sum $1^2 + 2^2 + \cdots n^2$. Let us call this sum S_n. We can then write the following.

$$\begin{aligned}(n+1)^3 - 1 &= 3S_n + \frac{3}{2}n(n+1) + n \\ \Rightarrow 3S_n &= (n+1)^3 - 1 - \frac{3}{2}n(n+1) - n \\ &= n^3 + 3n(n+1) + 1 - 1 - \frac{3}{2}n(n+1) - n \\ &= n^3 + 3n^2 + 3n - \frac{3}{2}n^2 - \frac{3}{2}n - n \\ &= n^3 + \frac{3}{2}n^2 + \frac{1}{2}n \\ &= \frac{1}{2}n(2n^2 + 3n + 1) \\ &= \frac{1}{2}n(2n^2 + 2n + n + 1) \\ &= \frac{1}{2}n(2n(n+1) + (n+1)) \\ &= \frac{1}{2}n(n+1)(2n+1) \\ \Rightarrow S_n &= \frac{1}{6}n(n+1)(2n+1) \end{aligned} \tag{2.24}$$

Algorithm 5 *The following algorithm computes the sum of the following series:*

$$1 \times 1 \times 1 + 2 \times 2 \times 1 + 2 \times 2 \times 1 + 2 \times 2 \times 2 + \cdots + n \times n \times n$$

```
sum = 0
for i = 1 to n do
   for j = 1 to i do
      for k = 1 to j do
         sum = sum + i * j * k
      endfor
   endfor
endfor
```

Once again, we need to have a size for the problem. Let the maximum value of an index, n, be the size of the problem. Let $T(n)$ be the time taken by the algorithm for a problem of size n. We write the expression for $T(n)$ below and simplify it. Let c_2 be the time taken by the work done inside the innermost loop. Let c_1 be the time taken by the algorithm outside all the loops. We sum the loop from inside.

$$
\begin{aligned}
T(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{j} c_2 + c_1 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{i} c_2 j + c_1 \\
&= c_2\sum_{i=1}^{n}\sum_{j=1}^{i} j + c_1 \\
&= c_2\sum_{i=1}^{n}\frac{1}{2}i(i+1) + c_1 \\
&= \frac{1}{2}c_2\sum_{i=1}^{n} i(i+1) + c_1 \\
&= \frac{1}{2}c_2\sum_{i=1}^{n}(i^2+i) + c_1 \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} i^2 + \sum_{i=1}^{n} i\right) + c_1 \\
&= \frac{1}{2}\left(\frac{1}{6}n(n+1)(2n+1) + \frac{1}{2}n(n+1)\right) + c_1 \\
&= \frac{1}{4}n(n+1)\left(\frac{1}{3}(2n+1)+1\right)c_2 + c_1 \\
&= \frac{1}{4}n(n+1)\left(\frac{2n+1+3}{3}\right)c_2 + c_1 \\
&= \frac{1}{12}n(n+1)(2n+4)c_2 + c_1
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{6}n(n+1)(n+2)c_2 + c_1 \\
&= \frac{1}{6}n(n^2+3n+2)c_2 + c_1 \\
&= \frac{1}{6}c_2n^2 + \frac{1}{2}c_2n^2 + \frac{1}{6}c_2n + c_1 \\
&= \Theta(n^3) \qquad (2.25)
\end{aligned}
$$

The polynomial expression we obtain for $T(n)$ has the highest degree of 3. We can ignore the smaller degree terms, and the coefficient of n^3, and write the result as $\Theta(n^3)$. This is a cubic algorithm in terms of time requirement.

2.4 Geometric Series

In a geometric series, the ratio between any two consecutive elements of the series is constant. The following is a general representation for a geometric series.

$$a, ar, ar^2, ar^3, \cdots \qquad (2.26)$$

The initial element is a. The ratio between two consecutive elements is r. The general or the ith element of the series is ar^{i-1}.

Once again, quite frequently, we have to obtain the sum of the first n elements of a geometric series. Therefore, S_n is given as below.

$$
\begin{aligned}
S_n &= a + ar + ar^2 + \cdots + ar^{n-1} \\
&= \sum_{i=1}^{n} ar^{i-1} \qquad (2.27)
\end{aligned}
$$

To obtain the sum S_n, we can proceed in the following manner. We write the sum in the expanded fashion.

$$S_n = a + ar + ar^2 + \cdots + ar^{n-1} \qquad (2.28)$$

We now multiply the whole equation, on the left hand side as well as on the right hand side, by r.

$$rS_n = ar + ar^2 + ar^3 + \cdots + ar^{n-1} + ar^n \qquad (2.29)$$

We write the two equations, one below the other. However, when writing the second equation, we write the terms on the right hand side, displaced by one term to the right. We then subtract the second equation from the first term by term.

$$
\begin{array}{rclcccccccccccc}
S_n &=& a & + & ar & + & ar^2 & + & \cdots & + & ar^{n-2} & + & ar^{n-1} & & \\
-rS_n &=& & - & ar & - & ar^2 & - & \cdots & - & ar^{n-2} & - & ar^{n-1} & - & ar^n \\
\hline
(1-r)S_n &=& a & & & & & & & & & & & - & ar^n \\
&=& a(1 & - & r^n) & & & & & & & & & &
\end{array}
$$

Therefore, we have

$$S_n = \frac{a(1-r^n)}{1-r} \tag{2.30}$$

We can alternatively write the following.

$$S_n = \frac{a(r^n-1)}{r-1} \tag{2.31}$$

The process of deriving S_n can be written in the following way also.

$$\begin{aligned}
S_n &= \sum_{i=1}^{n} ar^{i-1} \\
&= a + \sum_{i=2}^{n} ar^{i-1} \\
&= a + \sum_{i=2}^{n} ar^{i-1} + ar^n - ar^n \\
&= a + (\sum_{i=2}^{n} ar^{i-1} + ar^n) - ar^n \\
&= a + \sum_{i=2}^{n+1} ar^{i-1} - ar^n \\
&= a + r\sum_{i=1}^{n} ar^{i-1} - ar^n \\
&= a + rS_n - ar^n \\
S_n - rS_n &= a - ar^n \\
(1-r)S_n &= a(1-r^n) \\
S_n &= \frac{a(1-r^n)}{1-r}
\end{aligned} \tag{2.32}$$

Problem 9 *Obtain the sum of the following series:*

$$1 + 2 + 2^2 + \cdots + 2^k$$

Here,
a = the initial value = 1
r = the ratio between the two terms = 2
n = the number of terms = $k-1$
Therefore, S_n can be obtained as given below.

$$S_n = \frac{a(1-r^k)}{1-r}$$

$$
\begin{aligned}
&= \frac{1(1-2^k)}{1-2} \\
&= \frac{1-2^k}{-1} \\
&= 2^k - 1 \qquad (2.33)
\end{aligned}
$$

Problem 10 *Find the sum* $1+3+9+27+\cdots$ *to* 9 *terms.*

Let S be the sum we want to obtain. We can write

$$
\begin{aligned}
S &= 1+3+9+27+\cdots \\
&= 1+1\times 3+1\times 3^2+\cdots+1\times 3^8 \\
&= 1\times\frac{3^9-1}{2} \\
&= \frac{19683-1}{2} \\
&= 9841 \qquad (2.34)
\end{aligned}
$$

The second line shows that the series can written as a geometric series like the ones discussed earlier with initial element $a = 1$, and the ratio between two elements, $r = 3$. Hence, we can use the summation formula obtained for geometric series directly to obtain the answer.

2.4.1 A Special Case

When the absolute value of the ratio r between two consecutive terms is less than 1 , and the number of terms n is large, the formula can be simplified a bit. In such a case, we have an infinite series being summed. If $\mid r \mid< 1$, a compact sum can be found for such an infinite series. We need to take the limit of S_n as $n \to \infty$.

$$
\begin{aligned}
\lim_{n\to\infty} S_n &= \frac{a(1-r^n)}{1-r} \\
&= \frac{a}{1-r}\lim_{n\to\infty}(1-r^n) \\
&= \frac{a}{1-r}\left(1-\lim_{n\to\infty} r^n\right) \\
&= \frac{a}{1-r}(1-0) \\
&= \frac{a}{1-r} \qquad (2.35)
\end{aligned}
$$

Problem 11 *Find the sum of the following series:*

$$1, \frac{1}{2}, \frac{1}{2^2}, \cdots, \frac{1}{2^n}, \cdots$$

We want to obtain S_n given below.

$$S_n = 1 + \frac{1}{2} + \frac{1}{2^2} + \cdots \tag{2.36}$$

Here, $a = 1$ and $r = \frac{1}{2}$. Therefore, the sum is obtained as below.

$$\begin{aligned} S_n &= \frac{1}{1 - \frac{1}{2}} \\ &= \frac{1}{\frac{1}{2}} \\ &= 2 \end{aligned} \tag{2.37}$$

Therefore, the infinite series actually has a compact sum of 2.

2.5 Arithmetic-Geometric Series

A series where each element has two terms multiplied together, one term from a geometric series, and the other from an arithmetic series is called an *arithmetic-geometric series*. Let a general arithmetic series of n terms be written as $1, r, r^2, \cdots, r^{n-1}$ and a general geometric series of n terms be written as $a, a + d, a + 2d, \cdots, a + (n-1)d$. Now, if we take an element from the geometric series and the corresponding element from the arithmetic series and multiply them together, we get an arithmetic-geometric series that looks like:

$$1 \times a, r \times (a + d), \cdots, r^{n-1} \times [a + (n-1)]$$

$$\begin{array}{rclllllll} S &=& 1 \times a &+& r \times (a+d) &+ \cdots +& r^{n-1} \times [a + (n-1)d] & & \\ -rS &=& &-& r \times a &- \cdots -& r^{n-1} \times [a + (n-2)d] &-& r^n \times [a + (n-1)d] \\ \hline (1-r)S &=& a &+& rd &+ \cdots +& r^{n-1} d &-& r^n \times [a + (n-1)d] \\ &=& a & & rd & \times & \left(1 + r + r^2 + \cdots + r^{n-2}\right) &-& r^n \times [a + (n-1)d] \end{array}$$

From this, we can simplify to obtain the following.

$$\begin{aligned} (1-r)S &= a + rd \times \frac{r^{n-1} - 1}{r - 1} - r^n \times [a + (n-1)d] \\ \Rightarrow S &= \frac{a + rd \times \frac{r^{n-1}-1}{r-1} - r^n \times [a + (n-1)d]}{1 - r} \end{aligned} \tag{2.38}$$

Problem 12 *Sum the infinite series given below.*

$$1 + \frac{3}{5} + \frac{5}{5^2} + \frac{7}{5^3} + \cdots$$

Solution

$$\begin{array}{lcccccccl} S & = & 1 & + & \frac{3}{5} & + & \frac{5}{5^2} & + & \frac{7}{5^3} + \cdots \\ \frac{S}{5} & = & & & \frac{1}{5} & + & \frac{3}{5^2} & + & \frac{5}{5^3} + \frac{7}{5^4} + \cdots \\ - & = & & & - & & - & & - \\ \hline \frac{4S}{5} & = & 1 & + & \frac{2}{5} & + & \frac{2}{5^2} & + & \frac{2}{5^3} + \frac{2}{5^4} + \cdots \end{array}$$

From the above, we can get the sum S in the following manner.

$$\begin{aligned} \frac{4S}{5} &= 1 + \frac{2}{5}\left(1 + \frac{1}{5} + \frac{1}{5^2} + \cdots\right) \\ &= 1 + \frac{2}{5} \times \frac{1}{1 - \frac{1}{5}} \\ &= 1 + \frac{2}{5} \times \frac{1}{\frac{4}{5}} \\ &= 1 + \frac{2}{5} \times 54 \\ &= 1 + \frac{1}{2} \\ &= \frac{3}{2} \\ \Rightarrow S &= \frac{5 \times 3}{2 \times 4} \\ &= \frac{15}{8} && (2.39) \\ &= 1\frac{7}{8} && (2.40) \end{aligned}$$

Problem 13 *Sum the infinite series given below such that* $1 > x \geq 0$.

$$1 - 2x + 3x^2 - 4x^3 + 5x^4 - \cdots$$

Solution

Each term of the series can be composed of a term from a geometric series and one from an arithmetic series.

$$S = 1 + 2 \times (-x) + 3 \times (-x)^2 + 4 \times (-x)^3 + \cdots$$

Here, $1, 2, 3, \cdots$ are in an arithmetic series. $-x, (-x)^2, (-x)^3, \cdots$ are in a geometric series. To obtain the sum, we multiply the original series by the ratio $-x$ of the geometric series on both sides. We than subtract this multiplied series from the original series as given below.

$$\begin{array}{lcccccccccccc} S & = & 1 & - & 2x & + & 3x^2 & - & 4x^3 & + & 5x^4 & - & \cdots \\ -xS & = & & & x & - & 2x^2 & + & 3x^3 & - & 4x^4 & + & \cdots \\ \hline (1+x)S & = & 1 & - & x & + & x^2 & - & x^3 & + & x^4 & - & \cdots \end{array}$$

The expression on the right hand side of the inequality is a geometric series with the ratio $-x$ whose magnitude is less than 1. Therefore, we can now simplify this as follows.

$$\begin{aligned}(1+x)S &= \frac{1}{1+x} \\ \Rightarrow S &= \frac{1}{(1+x)^2}\end{aligned} \tag{2.41}$$

2.6 Harmonic Series

2.7 Exercises

Obtain the series sums given below.

1. $2+4+6+\cdots+2n$
2. $3+6+9+\cdots+3n$
3. $1+4+7+\cdots+(3n-2)$
4. $1+3+6+\cdots+\frac{1}{2}n(n+1)$
5. $2+6+18+\cdots+2\times 3^{n-1}$
6. $2^2+4^2+6^2+\cdots+(2n)^2$
7. $1^2+3^2+5^2+\cdots+(2n-1)^2$
8. $4^2+7^2+10^2+\cdots+(3n+1)^2$
9. $1^3+3^3+5^3+\cdots+(2n-1)^3$
10. $1\times 2+2\times 3+3\times 4+\cdots+n\times(n+1)$
11. $1\times 3+2\times 4+3\times 5+\cdots+n\times(n+2)$
12. $1\times 2+3\times 4+5\times 6+\cdots+(2n-1)\times(2n)$
13. $2\times 5+3\times 6+4\times 7+\cdots+(n+1)\times(n+4)$
14. $\frac{1}{1\times 2}+\frac{1}{2\times 3}+\frac{1}{3\times 4}+\cdots+\frac{1}{n(n+1)}$
15. $\frac{1}{1\times 3}+\frac{1}{3\times 5}+\frac{1}{5\times 7}+\cdots+\frac{1}{(2n-1)(2n+1)}$
16. $\frac{1}{1\times 4}+\frac{1}{4\times 7}+\frac{1}{7\times 10}+\cdots+\frac{1}{(3n-2)(3n+1)}$
17. $1\times 2\times 3+2\times 3\times 4+3\times 4\times 5+\cdots+n\times(n+1)\times(n+2)$
18. $1\times 2\times 3+2\times 3\times 5+3\times 4\times 7+\cdots+n\times(n+1)\times(2n+1)$

Chapter 3

Recurrence Relations

Recurrence relations occur quite frequently when analyzing algorithms. In this Chapter, we define a recurrence relation and present some examples. We also briefly introduce several techniques used for solving recurrence relations. There are various techniques for solving recurrence relations. Among these are the following.

- By repeated substitution,
- Using characteristic equations, and
- Using generating functions.

Finally, we solve a few recurrence relations using the technique of *Repeated Substitution*.

3.1 Recurrence Relations: Definitions and Examples

Assume we are given a sequence of numbers $< a_0, a_1, \cdots, a_n, \cdots >$. The angle brackets are used to include the entries in the sequence. The sequence can be finite or infinite.

A recurrence relation is an equation that relates a general term in the sequence, say a_n, with one of more preceding terms. The recurrence relation is said to be obeyed by the sequence. The following is a general recurrence relation.

$$a_n = f(a_{n-1}, a_{n-2}, \cdots, a_{n-m}) \qquad n \geq m \tag{3.1}$$

Here, the term a_n is a function of m immediately preceding terms of the sequence. If m is a small positive integer, a_n depends on a finite amount of history [GK90]. If $n = m$, the value of a_n depends on all the prior terms in the sequence and is called a *recurrence with full history*.

The following are examples of some general recurrence relations.

$$a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \cdots + c_m\, a_{n-m}$$

$$
\begin{aligned}
a_n &= c\, a_{n-1} + f(n) \qquad \text{f(n) is a function of n} \\
a_{n,m} &= a_{n-1,m} + a_{n-1,m-1} \\
a_n &= a_0\, a_{n-1} + a_1\, a_{n-2} + \cdots + a_{n-1}\, a_0
\end{aligned}
$$

Here, the first three are recurrence relations with finite history. The last one is a recurrence relation with full history.

When we solve recurrence relations in the study of algorithms, quite frequently, the relations are written in terms of a time or a space function, usually of just one argument—the size of the problem, written usually as n. The time function is usually written as $T(n), t(n), T_n$ or t_n. It represents the amount of time required by an algorithm to solve a problem of size n. For example, an algorithm can sort n elements, find the kth largest from among n elements, search for a target item in a list of n elements, etc. The following give examples of some simple recurrence relations that occur in the study of algorithms.

$$
\begin{aligned}
T(n) &= T(n-1) + c \\
T(n) &= T(n-1) + c\, n \\
T(n) &= T\left(\frac{n}{2}\right) + c \\
T(n) &= T\left(\frac{n}{2}\right) + c\, n \\
T(n) &= 2\, T\left(\frac{n}{2}\right) + c \\
T(n) &= 2\, T\left(\frac{n}{2}\right) + c\, n \\
T(n) &= 3\, T\left(\frac{n}{2}\right) + c \\
T(n) &= a\, T\left(\frac{n}{b}\right) + c\, n \\
T(n) &= T\left(\sqrt{n}\right) + c
\end{aligned}
$$

Each one of these recurrence relations, relates $T(n)$ with one prior term: $T(n-1)$, or $T\left(\frac{n}{2}\right)$, or $T\left(\frac{n}{b}\right)$, or $T\left(\sqrt{n}\right)$. So, each is of the form

$$T(n) = T(m) + f(n) \tag{3.2}$$

where $0 \le m < n$ and $f(n)$ is either a constant c or a function of n. Note that each recurrence relation must have one or more associated terminating cases of initial values. Terminating cases are not shown here.

Of course, more complex recurrences occur frequently in the study of algorithms. We start our discussion of recurrence relations with simple ones. Later in the book, we look at more complicated recurrences.

The space function is usually written as $S(n)$, $s(n)$, S_n or s_n. In the examples in this book, we use the time function although it can be easily replaced by the space function.

3.2 Solving Recurrences with Repeated Substitution

The technique of Repeated Substitution can be used to solve "simple" recurrence relations. The idea is very straight-forward. We start with the recurrence relation given to us. We use the recurrence relation to expand the right hand side of the equation. We do so a few times with the goal of finding a pattern on the right hand side as the argument becomes smaller. Once we find a pattern, we write a general expression on the right hand side. When we have a general expression, we can go ahead and solve the recurrence and obtain a closed-form solution. We illustrate the technique by solving a number of recurrences. In the course of performing this method, it is common practice to make assumptions regarding the values the argument n can take, in order to be able to solve a recurrence relation.

3.3 Solving the Recurrence $T(n) = T(n-1) + c$

The first recurrence relation we solve is given below.

$$T(n) = T(n-1) + c \qquad\qquad n > 1 \tag{3.3}$$

Here, c is a small positive constant.

We can illustrate the recurrence in the following manner. Assume $T(n)$ is the time required by an algorithm to solve a certain problem when the number of input elements is n. The algorithm takes c amount of time to remove one element from consideration, and then examines the $n-1$ elements left. This process is continued until a termination condition is reached.

Another way to understand recurrence relations is the following. The recurrence corresponds to an algorithm that makes one pass over each one of the n elements. It takes c time to examine an element.

3.3.1 Termination Condition

A recurrence relation needs one or more termination conditions to be solvable. The termination condition here is that when a sequence containing one element is given to the algorithm, the problem is trivially solvable in d amount of time. In terms of an equation, we can say the following.

$$T(1) = d \tag{3.4}$$

The two equations together give us a solvable recurrence.

3.3.2 Instantiations

We solve the recurrence by repeated substitution. In repeated substitution, we use the recurrence relation again and again, thereby reducing the size of the argument at each step. Each time we apply the original equation, we try to find a pattern among the reduced equations. Once we have found a pattern, we can write a general equation and obtain a tight sum. Thus,

$$T(n) = T(n-1) + c \tag{3.5}$$

is the recurrence relation we are going to use again and again. Assume n is an integer, $n \geq 1$. For example, if we want to instantiate the recurrence for an argument value of $n-1$, we get

$$T(n-1) = T(n-2) + c \tag{3.6}$$

Note that the argument to T on the right hand side is one less than the argument on the left hand side. Thus, if we instantiate the formula for an argument k, we get

$$T(k) = T(k-1) + c \tag{3.7}$$

Or, if we instantiate it for an argument value of 2, wet get

$$T(2) = T(1) + c \tag{3.8}$$

This is basically what we do in repeated substitution. We instantiate sequentially reducing the argument's value till we get to the termination situation. We usually add up all the instantiated equations, simplify what is left, perform some algebra, and obtain the solution we are interested in. The solution to the recurrence relation follows.

$$\begin{aligned} T(n) &= \boxed{\text{T (n-1) +c}} \\ &= (T(n-2)+c)+c \\ &= \boxed{\text{T (n-2) + 2c}} \\ &= (T(n-3)+c)+2c \\ &= \boxed{\text{T (n-c) + 3c}} \\ &\vdots \\ &= \boxed{\text{T (n-k) + kc}} \\ &\vdots \\ &= T(n-(n-1))+(n-1)c \\ &= T(1)+(n-1)c \\ &= T(1)+(n-1)c \\ &= d+(n-1)c \\ &= nc+(d-c) \end{aligned} \tag{3.9}$$

In the first line, we write the original equation. In the second line, we instantiate $T(n-1)$ using the original recurrence relation. In the third line, we simplify to obtain $T(n-2)+2c$. Next, we instantiate $T(n-2)$ using the original equation and simplify it to get $T(n-3)+3c$.

If we look at the three equations so far, we have the following.

$$T(n-1) + c$$
$$T(n-2) + 2c$$
$$T(n-3) + 3c$$

We see that the argument to T is n minus a certain number k, and whatever k is, it is the coefficient of c. As the argument to T comes down, the coefficient of c increases. After several such steps, we get to the general situation where if k is the number subtracted from n for the argument of T, the coefficient of c is $n-k$ as well. Therefore, the general form is

$$T(n-k) + kc \tag{3.10}$$

Finally, we reach the terminating situation when $n-1$ is subtracted from n as the argument of T. $n-(n-1) = 1$. Therefore, after the last expansion we do, we have

$$T(n-(n-1)) + (n-1)c \tag{3.11}$$

or

$$T(1) + (n-1)c \tag{3.12}$$

We know $T(1) = d$ from the terminating condition of the recurrence. Therefore, simplification produces $T(n) = d + (n-1)c$ as the solution.

3.4 Solving the Recurrence $T(n) = T(n-1) + cn$

We may have an algorithm that looks at all the elements of the input in every recursion and eliminates one element at a time. Several sorting algorithms are like this. For example, bubble sort looks at every element of input before it picks the smallest (or, largest) and lets it move to its destination.

The recurrence relation for such an algorithm can be written as

$$\begin{aligned} T(n) &= T(n-1) + cn \qquad n > 1 \\ &= d \end{aligned} \tag{3.13}$$

The first equation says that the algorithm looks at all n elements in the input. c is a small positive constant. The $T(n-1)$ term on the right hand side says there is one fewer element to look at in the next round. Note that the coefficient of $T(n-1)$ is also 1. d is also a small positive constant.

This recurrence relation can be solved using repeated substitution. The solution is given below.

$$T(n) = \boxed{\text{T (n-1) + cn}}$$

$$
\begin{aligned}
&= (T(n-2) + c(n-1)) + cn \\
&= \boxed{\text{T (n-2) + c((n-1) + n)}} \\
&= (T(n-3) + c(n-2)) + c((n-1) + n) \\
&= \boxed{\text{T (n-3) + c((n-2) + (n-1) + n)}} \\
&\ldots \\
&\ldots \\
&\ldots \\
&= T(n-k) + c((n-(k-1)) + \cdots + (n-1) + n) \\
&\ldots \\
&\ldots \\
&\ldots \\
&= T(n-(n-1)) + c((n-(n-2)) + \cdots + (n-1) + n) \\
&= T(1) + c(2 + 3 + \cdots + (n-1) + n) \\
&= d + c(1 + 2 + 3 + \cdots + (n-1) + n - 1) \\
&= d + c\left(\frac{1}{2}n(n+1) - 1\right) \\
&= \frac{1}{2}c\,n^2 + \frac{1}{2}c\,n + (d-c) \\
&= \Theta(n^2) \qquad (3.14)
\end{aligned}
$$

The first line is the original equation: $T(n-1) + cn$. After using the original recurrence relation to expand $T(n-1)$ and simplify a little, we get $T(n-2) + c((n-1) + n)$. After expanding $T(n-2)$ with the recurrence relation again and simplifying, we get $T(n-3) + c((n-2) + (n-1) + n)$. Now, we look at these carefully and try to discover a pattern to see how the expressions evolve. It is fairly easy to see an evolving pattern. We have a $T(\ )$ term with an argument $n-k$. Following the $T(\ )$ term, we have an arithmetic series multiplied by c. If the argument to $T(\ )$ is $n-k$, the arithmetic series starts with $n-(k-1) = n-k+1$. Therefore, the general term in the expansion is

$$T(n-k) + c((n-(k-1)) + \cdots + (n-1) + n).$$

In this example, we do not use the general expression, but it is something that is good to know.

We continue with the substitutions and get to the termination where the expansion looks like

$$T(1) + c(2 + 3 + \cdots + (n-1) + n).$$

In the second term, we multiply c by $2 + 3 + \cdots + (n-1) + n$. This is an arithmetic series starting with 2, and an increment of 1. We add 1 to the series and subtract 1 at the end for simplification. We know that $1 + 2 + \cdots + (n-1) + n$ adds up to $\frac{1}{2}n(n+1)$.

We perform additional algebraic manipulation to obtain

$$T(n) = \frac{1}{2}c\, n^2 + \frac{1}{2}c\, n + (d - c)$$

as the solution.

This solution can be written in asymptotic notation as $\Theta(n^2)$ by keeping the largest degree of n.

3.5 Solving the Recurrence $T(n) = T\left(\frac{n}{2}\right) + c$

In this section, we look at the recurrence relation corresponding to algorithm that behaves in the following fashion. The algorithm is given n elements to process. It takes a constant amount of time, here written as c to divide the original elements into two halves and decide which half to look at for further processing. This process of progressively dividing the elements into smaller and smaller halves continues till there is one element left in the half to be processed when the problem can be solved trivially.

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + c \qquad n > 1 \\ T(1) &= d \end{aligned} \tag{3.15}$$

Here, c and d are small positive constants.

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + c \\ &= \left[T\left(\frac{n}{2^2}\right) + c\right] + c \\ &= T\left(\frac{n}{2^2}\right) + 2c \\ &= \left[T\left(\frac{n}{2^3}\right) + c\right] + 2c \\ &= T\left(\frac{n}{2^3}\right) + 3c \\ &\vdots \\ &= T\left(\frac{n}{2^k}\right) + kc \\ &= T(1) + kc \end{aligned} \tag{3.16}$$

We are solving a recurrence relation, and we must assume that the recurrences terminate. For the recurrence to terminate, we make a simplifying assumption. We assume that n is a power of 2 to make the solution easy. In reality, it is unlikely that the number of original elements we are given is a perfect power of 2. However, in such cases, we can assume

that the number of elements we are dealing with is the next perfect power of 2. With this assumption, we will get an upper bound on the time consumed by the algorithm.
With the assumption that n is a perfect power of 2, we can write

$$n = 2^k \qquad\qquad k \geq 0$$

which leads us to the conclusion that

$$k = log_2 n$$

giving us the solution to the recurrence as

$$\begin{aligned} T(n) &= T(1) + kc \\ &= d + c\ log_2\ n \\ &= c\ log_2\ n + d \end{aligned} \tag{3.17}$$

In asymptotic terms, the solution can be written as $T(n) = log_2 n$. Such an algorithm is a lot faster than an algorithm that takes linear time.
In conclusion, we can say that an algorithm that takes constant time to divide up the original elements into two halves and decide which half to look at for the next iteration is a very efficient algorithm taking logarithmic time. An example of such an algorithm is the binary search algorithm wherein, during each pass of the algorithm we divide the elements into two halves and decide which half to look at by comparing the element being searched with the ”middle” element of the sorted sequence where we are looking. Note that the number of elements in the two halves in a real problem may not be equal. In other words, one ”half” may have one element more than the other ”half”. In the analysis above, we ignore this possibility and assume that the two halves are always equal.

3.6 Solving the Recurrence $T(n) = T\left(\frac{n}{2}\right) + c\ n$

The next recurrence we want to solve is the following.

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + c\ n \qquad\qquad n > 1 \\ T(1) &= d \end{aligned} \tag{3.18}$$

The recurrence results from a program that divides its input into two equal halves. To divide the input into two halves, the program goes over every element. That is why we have the linear factor $c\ n$. The initial condition says that when a single element is given to the algorithm, it takes a constant, d, amount of time.
We solve the recurrence using repeated substitution. The steps are given below.

$$\begin{aligned}
T(n) &= T\left(\frac{n}{2}\right) + c\,n \\
&= \left[T\left(\frac{n}{2^2}\right) + c\frac{n}{2}\right] + c\,n \\
&= T\left(\frac{n}{2^2}\right) + c\,n\left(\frac{1}{2} + 2\right) \\
&= \left[T\left(\frac{n}{2^3}\right) + c\,\frac{n}{2^2}\right] + c\,n\left(\frac{1}{2} + 1\right) \\
&= T\left(\frac{n}{2^3}\right) + c\,n\left(\frac{1}{2^2} + \frac{1}{2} + 1\right) \\
&\vdots
\end{aligned} \tag{3.19}$$

If we compare the boxed expressions, we begin to see a pattern. Following this pattern, we get the general expression as given below.

$$T(n) = T\left(\frac{n}{2^k}\right) + c\,n\left(\frac{1}{2^{k-1}} + \frac{1}{2^{k-2}} + \cdots + \frac{1}{2^2} + \frac{1}{2} + 1\right) \tag{3.20}$$

An important consideration needs to be discussed at this time. The n we start out with can be any positive integer. However, the steps we have shown above cannot be carried out for any value of n. Even, the first line $T(n) = T\left(\frac{n}{2}\right) + c\,n$ cannot be written unless n is an even number. If n is odd, one half contains $\lfloor \frac{n}{2} \rfloor$ elements, and the other half contains $\lceil \frac{n}{2} \rceil$ elements. Therefore, to be strictly correct, the recurrence should be the following.

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + c\,n \tag{3.21}$$

Here, $\lceil\ \rceil$ means ceiling and $\lfloor\ \rfloor$ means floor. We do not want to get into complications of $\lceil\ \rceil$s and $\lfloor\ \rfloor$s, and hence, make a simplifying assumption: for us to be able to write the recurrence relation n must be even, that is, divisible by 2. Then, we look at the repeated substitutions we have done so far, and conclude that for each substitution step, the current argument to $T(\ )$ must be divisible by 2. Thus, each of $\frac{n}{2}, \frac{n}{2^2}, \cdots, \frac{n}{2^{k-1}}$ must be divisible by 2. As a result, n is a power of 2. That is, we can assume $n = 2^k$ for some $k \geq 0$. In other words, we have to make the assumption that the n we start out with is a full power of 2. This is not realistic, but if the particular value of n we are looking at is not a power of 2, we can go up to the next full power of 2, and consider that to be the value of n. Recurrence problems like this one are not always solvable for all values of n. With this assumption, we can continue our solution. We know $n = 2^k$, and hence, $\frac{n}{2^k} = 1$.

With this background, we can continue with our solution to the recurrence. The summation inside the parentheses is a geometric series with a ratio of $\frac{1}{2}$ between any two consecutive terms.

$$\begin{aligned}
T(n) &= T(1) + c\,n\left(\frac{1}{2^{k-1}} + \frac{1}{2^{k-2}} + \cdots + \frac{1}{2^2} + \frac{1}{2} + 1\right) \\
&= d + c\,n\frac{1-\frac{1}{2^k}}{1-\frac{1}{2}} \\
&= d + 2\,c\,n\left(1 - \frac{1}{2^k}\right) \\
&= d + 2\,c\,n\left(1 - \frac{1}{n}\right) \\
&= d + 2\,c\,n - 2\,c \\
&= 2\,c\,n + (d - 2c) \qquad (3.22)
\end{aligned}$$

This is the solution to the recurrence. It says that if our algorithm considers only one half of the elements at a time, and it takes linear time to decide which half to look at, the whole algorithm takes linear time.

3.7 Solving the Recurrence $T(n) = 2\,T\left(\frac{n}{2}\right) + c$

In the previous sections, we have solved recurrence relations where we have $T(n)$ on the left hand side of the recurrence and $T\left(\frac{n}{2}\right)$ on the right hand side. The coefficient of $T\left(\frac{n}{2}\right)$ has been 1 in the two recurrences so far. In each one the recurrences, the solutions turned out to be linear functions of n. Let us now solve a recurrence where the coefficient of $T\left(\frac{n}{2}\right)$ is 2. Later we will consider recurrences where the coefficient of $T\left(\frac{n}{2}\right)$ is more than 2.

The discussions here can be generalized. A general way to look at the recurrences we have solved so far is that the initial problem has a problem of size n. In other words, the original problem has n elements (e.g., numbers) to deal with. The algorithm divides up the original n elements into b parts, each of equal size in the ideal case. In the recurrences solved in the previous two sections the value of b is 2, but in general, it can be some other integer, say 3. For our current discussions, let us assume that b is 2 as well. The algorithm then decides to look at one of the b parts for further processing. To divide up the elements into two halves and decide which half to look at, the algorithm takes either constant or linear time. Now, in this section, we will see what happens when the algorithm has to look at more than one half in subsequent processing.

This recurrence relation arises in a recursive program or algorithm that makes a linear pass through the input, before, during or after it is split into two halves. Both halves are looked at or processed. However, unlike the recurrence solved in the previous section, it takes only a fixed amount of time to decide which half to look at for the next pass of the algorithm.

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + c \qquad n > 1 \\
T(1) &= d \qquad (3.23)
\end{aligned}$$

The steps are similar to the recurrence relations we have solved earlier. We repeat the process below for this specific problem.

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + c \\
&= 2\left[2T\left(\frac{n}{2^2}\right) + c\right] + c \\
&= 2^2 T\left(\frac{n}{2^2}\right) + (2+1)c \\
&= 2^2\left[2T\left(\frac{n}{2^3}\right) + c\right] + (2+1)\,c \\
&= 2^3 T\left(\frac{n}{2^3}\right) + \left(2^2+2+1\right)c \\
&\vdots \\
&= 2^k T\left(\frac{n}{2^k}\right) + \left(2^{k-1} + 2^{k-2} + \cdots + 2^2 + 2 + 1\right)c \\
&= nT(1) + \frac{2^k - 1}{2-1}c \\
&= nd + \left(2^k - 1\right)c \\
&= nd + (n-1)c \\
&= (c+d)n - c \qquad (3.24)
\end{aligned}$$

The first line is the original recurrence. In the second line, we use the recurrence again to obtain an expansion $T\left(\frac{n}{2}\right)$ as $2T\left(\frac{n}{2^2}\right) + c$. We simplify to obtain the third line. Next, we use the recurrence again to obtain an expansion for $T\left(\frac{n}{2^2}\right)$ as $2T\left(\frac{n}{2^3}\right) + c$. After some simplifying algebra, we obtain the fifth line in the derivation.

If we look at the first, third and the fifth lines in the expansion above and compare the progression from one to the next, we see a clear pattern emerging. The pattern has two parts that are added together. The first part is of the form $2^k T\left(\frac{n}{2^k}\right)$ for an arbitrary positive integer value of k. The second part is a geometric series $\left(2^{k-1} + 2^{k-2} + \cdots + 2^2 + 2 + 1\right)$ for the same value of k. To terminate the repeated substitutions, we make the simplifying assumption that n, the number of initial elements we are dealing with is a power of 2, and that in particular, $n = 2^k$ for a certain value of $k \geq 1$. The justification for this assumption has been discussed in the previous section. With the assumption $n = 2^k$, $T\left(\frac{n}{2^k}\right)$ can be replaced by $T(1)$. We know $T(1) = d$ from the initial specification of the recurrence's termination condition.

The second part of the general pattern contains $\left(2^{k-1} + 2^{k-2} + \cdots + 2^2 + 2 + 1\right)$ which is a geometric series with a ratio of 2 and it adds up to $2^k - 1$. After simplification, we obtain that $T(n) = (c+d)n - c$ which is a linear function of n. This result can be written as $\Theta(n)$ in asymptotic notation.

3.8 Solving the Recurrence $T(n) = 2T\left(\frac{n}{2}\right) + cn$

The next recurrence we want to solve is the following.

$$\begin{aligned} T(n) &= 2\,T\left(\frac{n}{2}\right) + c\,n \qquad\qquad n > 1 \\ T(1) &= d \end{aligned} \tag{3.25}$$

The recurrence results from a program that divides its input into two equal halves. The program solves the problem for both halves. To divide the input into two halves and/or to be able to put the results of solving t he two halves back, the program goes over every element. That is why we have the linear factor $c\ n$. The initial condition says that when a single element is given to the algorithm, it takes a constant, d, amount of time.

The steps in the solution of the recurrence are given below.

$$\begin{aligned} T(n) &= 2\,T\left(\frac{n}{2}\right) + cn \\ &= 2\left[2\,T\left(\frac{n}{2^2}\right) + c\frac{n}{2}\right] + cn \\ &= 2^2\,T\left(\frac{n}{2^2}\right) + cn + cn \\ &= 2^2\,T\left(\frac{n}{2^2}\right) + 2cn \\ &= 2^2\left[2\,T\left(\frac{n}{2^3}\right) + c\,\frac{n}{2^2}\right] + 2cn \\ &= 2^3T\left(\frac{n}{2^3}\right) + cn + 2cn \\ &= 2^3T\left(\frac{n}{2^3}\right) + 3cn \\ &\vdots \\ &= 2^k\,T\left(\frac{n}{2^k}\right) + k\ c\ n \\ &= 2^k\,T(1) + k\ c\ n \\ &= nd + c\ n\ log_2\ n \\ &= \Theta(n\ log_2\ n) \end{aligned} \tag{3.26}$$

The first line above is the original recurrence. In the second line, we use the original recurrence to expand $T\left(\frac{n}{2}\right)$ as $2\ T\left(\frac{n}{2^2}\right) + c\frac{n}{2}$. The third and the fourth lines are simple algebraic simplifications. In the fifth line of the derivation above, we expand $T\left(\frac{n}{2^2}\right)$ as $2\,T\left(\frac{n}{2^3}\right) + c\,\frac{n}{2^2}$. The next two lines are simple algebraic simplifications. If we compare the expressions on the right hand side of the equality in lines 1, 4 and 7, we see a clear pattern of expansion emerging. This expansion leads to the general expansion $2^k\;T\left(\frac{n}{2^k}\right) + kcn$ below the three vertical dots for some positive integer value of $k \geq 1$. To finish up the continued expansion of the original expression, we make the simplifying assumption that $n = 2^k, k \geq 1$, just like we have done in the previous sections. With this assumption, we conclude that $\frac{n}{2^k} = 1$. Thus, $T\left(\frac{n}{2^k}\right) = T(1) = d$. Since, $2^k = 1$ by our simplifying assumption, the first term in the final solution becomes nd. The second part of the general expression is kcn. Since, $n = 2^k$ by our assumption, $k = log_2\,n$. This leads us to write the second term as $c\;n\;log_2\;n$. Therefore, the final solution to the recurrence is $nd + c\;n\;log_2\;n$. In this solution, the first term is linear in n since d is a small positive constant. In the second term, c is a small positive constant. Thus, the second term $c\;n\;log_2\;n$ increases at a faster rate than the first term as n becomes large. Therefore, in an asymptotic specification, we can write the solution as $\Theta(n\;log_2\;n)$.

3.9 Solving the Recurrence $T(n) = 3\,T\left(\frac{n}{2}\right) + c$

The next recurrence we want to solve is the following.

$$\begin{aligned} T(n) &= 3\,T\left(\frac{n}{2}\right) + c \qquad\qquad n > 1 \\ T(1) &= d \end{aligned} \tag{3.27}$$

We solve the recurrence by repeated substitution, just like the recurrences we have solved before.

$$\begin{aligned} T(n) &= 3\,T\left(\frac{n}{2}\right) + c \\ &= 3\left[3\,T\left(\frac{n}{2^2}\right) + c\right] + c \\ &= 3^2\,T\left(\frac{n}{2^2}\right) + (3+1)c \\ &= 3^2\left[3\,T\left(\frac{n}{2^3}\right) + c\right] + (3+1)c \\ &= 3^3\,T\left(\frac{n}{2^3}\right) + \left(3^2+3+1\right)c \\ &\vdots \\ &= 3^k\,T\left(\frac{n}{2^k}\right) + \left(3^{k-1} + 3^{k-2} + \cdots + 3 + 1\right)c \\ &= 3^k\,T(1) + c\,\frac{3^k-1}{3-1} \end{aligned}$$

$$
\begin{aligned}
&= 3^k\ d + \frac{1}{2}c\left(3^k - 1\right) \\
&= \left(d + \frac{1}{2}\ c\right) 3^k - \frac{1}{2}\ c \\
&= b\ 3^k - \frac{1}{2}\ c \\
&= b\ n^{1.58496} - \frac{1}{2}c \qquad (3.28)
\end{aligned}
$$

The first line is the original recurrence. In the second line, we expand $T\left(\frac{n}{2}\right)$ by $3\ T\left(\frac{n}{2^2}\right) + c$ using the recurrence. The third line is obtained after algebraic simplification. In the fourth line, we replace $T\left(\frac{n}{2^2}\right)$ by $3\ \ T\left(\frac{n}{2^3}\right) + c$ using the original recurrence. The fifth line is obtained after algebraic simplification. Now, if we compare the expressions on the right hand side of the equality in lines 1, 3 and 5, we see a pattern that is beginning to emerge. This pattern is expressed in terms of a variable k in the line following the vertical dots. This is the general pattern. Here, we have performed repeated substitution $k-1$ number of times. To finish our derivation, we assume that n is a power of k. In particular, we assume that $n = 2^k, k \geq 1$. The general expression in the line following the vertical dots has two parts. The first part is $3^k\ T\left(\frac{n}{2^k}\right)$. Since, we assume $n = 2^k$, we can rewrite this as $3^k\ \ T(1)$. From the termination condition of the recurrence, $T(1) = d$. Therefore, the first part of the general expression becomes $3^k\ \ d$. The second part of the general expression in the line following the vertical dots is $\left(3^{k-1} + 3^{k-2} + \cdots + 3 + 1\right) c$. The part inside the parentheses is a geometric series with ratio 3 between consecutive items. The geometric series can be summed as $frac3^k - 13 - 1$ which simplifies to $\frac{1}{2}\left(3^k - 1\right)$. After this, we perform some algebraic simplification to obtain the line $b\ 3^k - \frac{1}{2}\ c$.

Now, we need to obtain an expression for 3^k in terms of a power of n. This is done in the following manner. We made the assumption $n = 2^k$. This gives

$$
\begin{aligned}
k &= log_2\ n \\
&= log_3\ n\ \ log_2\ 3
\end{aligned}
$$

after we change the base of the logarithm. Further, we continue as the following to obtain a value for 3^k in terms of a power of n.

$$
\begin{aligned}
3^k &= 3^{log_3\ n\ log_2\ 3} \\
&= \left(3^{log_3\ n}\right)^{log_2\ 3} \\
&= n^{log_2\ 3} \\
&= n^{1.58496}
\end{aligned}
$$

This is the expression we use instead of 3^k in solving the recurrence above.

Our solution to the recurrence in this section shows that if the coefficient of $T\left(\frac{n}{2}\right)$ becomes higher than 2, the resulting solution of the recurrence becomes non-linear. That is, the power of n in the solution of the recurrence becomes more than 1.

The final solution can be written in asymptotic notation as

$$T(n) = \Theta\left(n^{1.58496}\right)$$

Here, as we have just observed the power of n is $log_2\ 3 = 1.58496$.
One can next try to solve another recurrence where the coefficient of $T\left(\frac{n}{2}\right)$ is 4 instead of 3 in the recurrence we solved. That is, if we solve the recurrence,

$$\begin{aligned} T(n) &= 4\,T\left(\frac{n}{2}\right) + c \\ T(1) &= d \end{aligned} \tag{3.29}$$

we will see that the solution will come out as

$$T(n) = \Theta(n^2)$$

where the power of n is $log_2\ 4 = 2$. In general, if we solve

$$\begin{aligned} T(n) &= a\,T\left(\frac{n}{b}\right) + c & n \geq 1 \\ T(1) &= d \end{aligned} \tag{3.30}$$

where a and b are positive integers, and $a > b$, the solution to the recurrence will come out as

$$T(n) = n^{log_b\ a} \tag{3.31}$$

This is a general theorem and will be discussed later in the chapter.

3.10 Solving the Recurrence $T(n) = T\left(\sqrt{n}\right) + c$

The recurrence we want to solve now is

$$\begin{aligned} T(n) &= T(\sqrt{n}) + c & n > 2 \\ &= d & n = 1, 2 \end{aligned} \tag{3.32}$$

This recurrence is different from the ones we have seen so far in that in each recursion, the number of elements to look at is reduced to a square root of the number of elements in the previous step.

$$\begin{aligned} T(n) &= T\left(\sqrt{n}\right) + c \\ &= T\left(n^{\frac{1}{2}}\right) + c \\ &= \left[T\left(n^{\frac{1}{2^2}}\right) + c\right] + c \end{aligned}$$

$$
\begin{aligned}
&= T\left(n^{\frac{1}{2^2}}\right) + 2c \\
&= \left[T\left(n^{\frac{1}{2^3}}\right) + c\right] + 2c \\
&= T\left(n^{\frac{1}{2^3}}\right) + 3c \\
&\vdots \\
&= T\left(n^{\frac{1}{2^k}}\right) + kc \\
&= T(2) + kc \\
&= d + c\, log_2\, log_2\, n \\
&= \Theta\left(log_2\, log_2\, n\right) \qquad (3.33)
\end{aligned}
$$

The first line is the original recurrence. The second line is a rewrite of the first with the square root replaced by a power of $\frac{1}{2}$. In the third line, we replace $T\left(n^{\frac{1}{2}}\right)$ by its expansion using the recurrence relation we are trying to solve. We rewrite as line four. Next, we rewrite $T\left(n^{\frac{1}{2^2}}\right)$ as $T\left(n^{\frac{1}{2^3}}\right) + c$ and simplify to obtain $T\left(n^{\frac{1}{2^3}}\right) + 3c$. If we compare lines 2, 4 and 6, we see a pattern emerging and the general pattern is written as $T\left(n^{\frac{1}{2^k}}\right) + kc$. We need to simplify this to obtain a solution to the recurrence. For this, we make the assumption that $n^{\frac{1}{2^k}} = 2$. In other words,

$$
\begin{aligned}
n^{\frac{1}{2^k}} &= 2 \\
\Rightarrow \sqrt[2^k]{n} &= 2 \\
\Rightarrow n &= 2^{2^k} \\
\Rightarrow 2^k &= log_2\, n \\
\Rightarrow k &= log_2\, log_2\, n
\end{aligned}
$$

Because, $k = log_2\ log_2\ n$, we can write the solution to the original recurrence as $T(n) = T(2) + kc$ which can be simplified as $T(n) = d + c\ log_2\ log_2\ n$. This can be written asymptotically as $T(n) = \Theta\left(log_2\ log_2\ n\right)$.

3.11 Solving the Recurrence $T(n) = 2\ T\left(\sqrt{n}\right) + log_2\ n$

3.11.1 Solving Using Repeated Substitution From First Principles

This is a fairly complex recurrence, but it can solved using repeated substitution just like the other recurrences we have solved in this chapter.

The recurrence to solve is

$$
\begin{aligned}
T(n) &= 2\ T\left(\sqrt{n}\right) + log_2\ n \qquad n > 1 \\
T(2) &= d \qquad (3.34)
\end{aligned}
$$

The solution steps are given below.

$$\begin{aligned}
T(n) &= 2\,T\left(n^{\frac{1}{2}}\right) + log_2\ n \\
&= 2\left[2\,T\left(n^{\frac{1}{2^2}}\right) + log_2\ n^{\frac{1}{2}}\right] + log_2\ n \\
&= 2^2\ T\left(n^{\frac{1}{2^2}}\right) + 2\ log_2\ n^{\frac{1}{2}} + log_2\ n \\
&= 2^2\ T\left(n^{\frac{1}{2^2}}\right) + 2 \times \frac{1}{2}\ log_2\ n + log_2\ n \\
&= 2^2\ T\left(n^{\frac{1}{2^2}}\right) + 2\ log_2\ n \\
&= 2^2\left[2\,T\left(n^{\frac{1}{2^3}}\right) + log_2\ n^{\frac{1}{2^2}}\right] + 2\ log_2\ n \\
&= 2^3\ T\left(n^{\frac{1}{2^3}}\right) + 2^2\ log_2\ n^{\frac{1}{2^2}} + 2\ log_2\ n \\
&= 2^3\ T\left(n^{\frac{1}{2^3}}\right) + 2^2 \times \frac{1}{2^2}\ log_2\ n + 2\ log_2\ n \\
&= 2^3\ T\left(n^{\frac{1}{2^3}}\right) + 3\ log_2\ n \\
&\vdots \\
&= 2^k\ T(2) + k\ log_2\ n \\
&= log_2\ n + log_2\ n\ \ log_2\ log_2\ n \qquad (3.35)
\end{aligned}$$

Just like the previous recurrence, for simplicity let us assume

$$\begin{aligned}
n^{\frac{1}{2^k}} &= 2 \\
\Rightarrow \sqrt[2^k]{n} &= 2 \\
\Rightarrow n &= 2^{2^k} \\
\Rightarrow 2^k &= log_2\ n \\
\Rightarrow k &= log_2\ log_2\ n
\end{aligned}$$

This allows us to write the last line of the derivation above.

3.11.2 Solving Using Variable Substitution

Some recurrence relations can be solved easily by making up new variables that can then be used to simplify the look of the recurrence, and thus its solution. One such recurrence that can be solved using variable substitution is the one being discussed.

$$T(n) = 2\,T\left(\sqrt{n}\right) + log_2\ n \qquad (3.36)$$

To simplify matters, let us introduce a variable m so that $n = 2^m$. With this assumption, we can rewrite the recurrence above as

$$T\left(2^m\right) = 2\,T\left(2^{\frac{m}{2}}\right) + m \qquad (3.37)$$

Let us now introduce a new function T' such that $T'(m) = T\left(2^m\right)$. This leads to

$$S(m) = 2\ S\left(\frac{m}{2}\right) + m \tag{3.38}$$

This is a familiar recurrence now. Its solution was obtained in Section... The solution is

$$\begin{aligned} S(m) &= m\ log_2\, m + c \\ \Rightarrow T\left(2^m\right) = m\ log_2\, m + c & \\ \Rightarrow T(n) = log_2\, n\ log_2\, log_2\, n + c & \end{aligned} \tag{3.39}$$

3.12 Solving the Recurrence $T(n) = \frac{2}{n}\sum_{i=0}^{n-1} T(i) + cn$

This recurrence occurs when we analyze the behavior of a sorting algorithm called *Quicksort*. It does not look like a recurrence that can be solved easily. It looks much more complex than the recurrences we have solved so far because this recurrence has a summation of $T()$s on the right hand side. However, with some algebra, this recurrence can be converted to one we can solve using repeated substitution. Note that we are given a termination condition:

$$T(0) = d \tag{3.40}$$

for some positive constant d.

The first thing we do is go through some steps to rewrite this recurrence. The original recurrence is given below.

$$T(n) = \frac{2}{n}\sum_{i=0}^{n-1} T(i) + cn \tag{3.41}$$

Multiplying throughout by n, we get the following.

$$n\ \ T(n) = 2\sum_{i=0}^{n-1} T(i) + cn^2 \tag{3.42}$$

If we instantiate Equation 3.42 equation for the argument value $n-1$ of T, we get

$$(n-1)\ \ T(n-1) = 2\sum_{i=0}^{n-2} T(i) + c(n-1)^2 \tag{3.43}$$

We subtract Equation 3.43 from Equation 3.42 to obtain the difference. Next, we simplify the difference by performing some algebra.

$$\begin{aligned} n\ \ T(n) - (n-1)\ \ T(n-1) &= \left[2\ \ \sum_{i=0}^{n-1} T(i) + cn^2\right] - \left[2\ \ \sum_{i=0}^{n-2} T(i) + c(n-1)^2\right] \\ &= 2\ \ T(n-1) + c\left[n^2 - (n-1)^2\right] \\ &= 2\ \ T(n-1) + c(2n-1) \end{aligned} \tag{3.44}$$

Note that $\sum_{i=0}^{n-1} T(i)$ is a short form for writing $T(0) + T(1) + \cdots + T(n-2) + T(n-1)$, and $\sum_{i=0}^{n-2} T(i)$ is a short form for writing $T(0) + T(1) + \cdots + T(n-3) + T(n-2)$. Thus, subtracting $\sum_{i=0}^{n-2} T(i)$ from $\sum_{i=0}^{n-1} T(i)$ leaves only one term $T(n-1)$. Next, we move the $(n-1)\ \ T(n-1)$ term to the right hand side, and then divide throughout by $n(n+1)$ to get the following.

$$\begin{aligned} nT(n) &= (n+1)T(n-1) + c(2n-1) \\ \Rightarrow \frac{T(n)}{n+1} &= \frac{T(n-1)}{n} + \frac{c(2n-1)}{n(n+1)} \end{aligned} \tag{3.45}$$

We will use Equation 3.45 for the method of Repeated Subsitution in order to solve for $T(n)$. But, before proceed with repeated substitution, it makes sense for us to write $\frac{c(2n-1)}{n(n+1)}$ in terms of partial fractions. Partial fractions have been discussed in Section 1.2 of this book. We can easily show that

$$\frac{c(2n-1)}{n(n+1)} = -\frac{1}{n} + \frac{3}{n+1}. \tag{3.46}$$

Using these partial fractions, we will rewrite Equation 3.45 and continue with solution by repeated substitution. We keeep $\frac{T(n)}{n+1}$ instead of $T(n)$ on the left hand side in the repeated substitutions because it makes the algebraic manipulations simpler. Thus, the recurrence relation we solve by repeated substitution is the one given below.

$$\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + c\left[-\frac{1}{n} + \frac{3}{n+1}\right] \tag{3.47}$$

The solution of the recurrence relation Equation 3.47 follows.

$$\begin{aligned} \frac{T(n)}{n+1} &= \frac{T(n-1)}{n} + c\left[-\frac{1}{n} + \frac{3}{n+1}\right] \\ &= \left\{\frac{T(n-2)}{n-1} + c\left[-\frac{1}{n-1} + \frac{3}{n}\right]\right\} + c\left[-\frac{1}{n} + \frac{3}{n+1}\right] \\ &= \frac{T(n-2)}{n-1} - c\left[\frac{1}{n-1} + \frac{1}{n}\right] + 3c\left[\frac{1}{n} + \frac{1}{n+1}\right] \\ &= \left\{\frac{T(n-3)}{n-2} + c\left[-\frac{1}{n-2} + \frac{3}{n-1}\right]\right\} - c\left[\frac{1}{n-1} + \frac{1}{n}\right] + 3c\left[\frac{1}{n} + \frac{1}{n+1}\right] \\ &= \frac{T(n-3)}{n-2} - c\left[\frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n}\right] + 3c\left[\frac{1}{n-1} + \frac{1}{n} + \frac{1}{n+1}\right] \\ &\ \vdots \\ &= \frac{T(0)}{1} - c\left[\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}\right] + 3c\left[\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n+1}\right] \\ &= d - cH_n + 3c\left[\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right] - 3c + \frac{3c}{n+1} \end{aligned}$$

$$
\begin{aligned}
&= d - cH_n + 3cH_n - 3c + \frac{3c}{n+1} \\
&= d + 2cH_n - 3c + \frac{3c}{n+1} \qquad (3.48)
\end{aligned}
$$

The first line above is a rewrite of Equation 3.45 where we replace the more complex fraction with its partial fractions. In the second line, we obtain a substitution for $\frac{T(n-1)}{n}$ using Equation 3.47. The expansion for $\frac{T(n-1)}{n}$ is given within big curly braces ({ and }) in the second line. Next, we obtain an expansion for $\frac{T(n-2)}{n-1}$, once again using Equation 3.47. Then, we perform the algebra to write the negative and positive parts separately.

If we look at the first, the third and the fifth lines of the expansion above, obtained by repeated substitution, we see a pattern emerging. The general expansion has three terms added together: the first is of the form $\frac{T(n-k-1)}{n-k}$ for a positive integer k less than or equal to n, the second is $-c\left[\frac{1}{n-k} + \cdots \frac{1}{n-1} + \frac{1}{n}\right]$, and the third is $3c\left[\frac{1}{n-k+1} + \cdots + \frac{1}{n+1}\right]$. If we continue with the repeated substitutions till we can perform no more substitution, we end up with the line below the three vertical dots:

$$
\frac{T(0)}{1} - c\left[\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}\right] + 3c\left[\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n+1}\right].
$$

We know that the value of $T(0)$ is d from the termination condition 3.40. Also, the value of $\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ is usually written as H_n. It is the sum of the terms of the Harmonic series: $1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}$. We perform some algebraic manipulations to obtain 3.48.

We can now multiply both sides of Equation 3.48 by $n+1$ to get

$$
\begin{aligned}
T(n) &= (n+1)d + 2c(n+1)H_n - 3c(n+1) + 1 \\
&= (n+1)d + 2c(n+1)ln\ n - 3c(n+1) + 1 \\
&= \Theta(n\ ln\ n) \\
&= \Theta(0.69344\ n\ log_2\ n) \\
&= \Theta(n\ log_2\ n) \qquad (3.49)
\end{aligned}
$$

Note that mathematicians spent many years centuries ago studying H_n in general, and an accepted value for H_n is

$$
H_n \approx ln\ n + \gamma \qquad (3.50)
$$

where γ is the Euler constant and has a value of 0.57721 approximately. Here, the base of the logarithm is e whose value is 2.71828 approximately. Therefore,

$$
\begin{aligned}
ln\ \ n &= log_e\ \ n \\
&= log_e\ 2\ \ log_2\ n \\
&= 0.69314\ \ log_2\ n
\end{aligned}
$$

As a result, we can write the final answer as $\Theta(n\ log_2\ n)$ in asymptotic notation.

3.13 Exercises

Solve the following recurrence relations. If necessary make up appropriate termination conditions.

1. $T(n) = 2\,T(n-1) + 5$
2. $T(n) = 3\,T(n-1)$
3. $T(n) = T\left(\frac{n}{2}\right) + c\ log\ n$
4. $T(n) = T\left(\frac{n}{2}\right) + c\,n^2$
5. $T(n) = 2\,T\left(\frac{n}{2}\right) + log\ n$
6. $T(n) = 8\,T\left(\frac{n}{2}\right) + n^2$
7. $T(n) = 2\,T\left(\frac{n}{2}\right) + n^3$
8. $T(n) = 2\,T\left(\frac{9n}{10}\right) + n$
9. $T(n) = 16\,T\left(\frac{n}{2}\right) + (n\ log\ n)^4$
10. $T(n) = 7\,T\left(\frac{n}{3}\right) + n$
11. $T(n) = 9\,T\left(\frac{n}{3}\right) + n^3\ log\ n$
12. $T(n) = 2\,T\left(\frac{n}{4}\right) + \sqrt{n}$
13. $T(n) = 3\,T\left(\frac{n}{2}\right) + n\ log\ n$
14. $T(n) = 5\,T\left(\frac{n}{5}\right) + \frac{n}{log\ n}$
15. $T(n) = 4\,T\left(\frac{n}{2}\right) + n^2\ \sqrt{n}$
16. $T(n) = 2\,T\left(\frac{n}{2}\right) + \frac{n}{log\ n}$
17. $T(n) = T(n-1) + \frac{1}{n}$
18. $T(n) = T(n-1) + log\ n$
19. $T(n) = T(n-2) + 2\ log\ n$
20. $T(n) = \sqrt{n}\ T(\sqrt{n}) + n$

Chapter 4

Characteristic Equations

A recurrence relation is a relation among different terms in a sequence. In Chapter 3, we discussed how simple recurrence relations can be solved by the method of Repeated Substitution. We worked out a large number of examples in that chapter. In this chapter, we will still deal with simple recurrences, but of a specific type called linear recurrences with constant coefficients.

In this chapter, we use t_n and $T(n)$ interchangeably to denote the nth term of a series. Consider a recurrence of the form

$$a_0 \, t_n + a_1 \, t_{n-1} + \cdots + a_k \, t_{n-k} = 0 \tag{4.1}$$

where k and a_i terms are constants. Such an equation is called a *homogeneous linear recurrence equation with constant coefficients*. The coefficients of the t_i terms, i.e., the a_i terms are *constants*. The t_i terms are linear, i.e., there are no products of t_i terms or squares or cubes, etc. It is called *homogeneous* because the right hand side is 0. Such a recurrence relation can be solved by Repeated Substitution, but sometimes it may become too tedious, or it may be difficult to see the emerging pattern. It is easy to solve such a recurrence relation by using a technique that uses *characteristic equations*. We discuss this method in this chapter.

When we look at solutions to such recurrence relations first, we make the assumption that each root of the characteristic equation is unique. This is the simplest case. Then, we expand on this simple case assuming that a root can occur several times, i.e., we have so called *mulitple roots*.

If the right hand side of Equation 4.1 is not zero or the zero function, it is called a *non-homogeneous linear recurrence equation with constant coefficients*. Thus, if $f(n) \neq 0$,

$$a_0 \, t_n + a_1 \, t_{n-1} + \cdots + a_k \, t_{n-k} = f(n) \tag{4.2}$$

is a non-homogeneous linear recurrence equation with constant coefficients. There is no general methods for solving a non-homogeneous linear recurrence equation although we can solve many complex recurrence relations using techniques such as Generating Functions discussed in Chapter 5. However, for some simple types of $f(n)$ functions, Equation

4.2 can be easily solved as as variations of Equation 4.1. A common special case of Equation 4.2 that can be solved easily is given below.

$$a_0\, t_n + a_1\, t_{n-1} + \cdots + a_k\, t_{n-k} = b^n\, p(n). \tag{4.3}$$

Here b is a constant and $p(n)$ is a polynomial. The solution turns out to be similar to solutions to Equation 4.1 with multiple roots.

When we rewrite the recurrence relations listed in Section 3.1 by moving the t_i terms to the left hand side, we get the following.

$$\begin{aligned}
t_n - t_{n-1} &= c \\
t_n - t_{n-1} &= cn \\
t_n - t_{\frac{n}{2}} &= c \\
t_n - t_{\frac{n}{2}} &= cn \\
t_n - 2\, t_{\frac{n}{2}} &= c \\
t_n - 2\, t_{\frac{n}{2}} &= cn \\
t_n - 3\, t_{\frac{n}{2}} &= c \\
t_n - a\, t_{\frac{n}{b}} &= cn \\
t_n - 2\, t_{\sqrt{n}} &= c
\end{aligned}$$

Note that we have written t_i instead $T(i)$ everywhere.

Clearly, only the first two are linear recurrence relations with constant coefficients. The rest are not linear because they have terms such as $\frac{n}{2}, t_{\frac{n}{b}}, t_{\sqrt{n}}$. There are other linear recurrence relations with constant coefficients that occur in algorithms analysis. One common example is the recurrence relation needed to define the nth Fibonacci number:

$$t_n = t_{n-1} + t_{n-2} \qquad\qquad n \geq 2$$

or,

$$t_n - t_{n-1} - t_{n-2} = 0 \qquad\qquad n \geq 2.$$

This is a recurrence that is very easy to solve using the method of characteristic equations, but quite difficult if we want to use the method of Repeated Substitutions.

4.1 Solving Homogeneous Linear Recurrences

Let us first start by looking at the following recurrence relations.

$$\begin{aligned}
t_n &= 4\, t_{n-1} \\
t_n &= t_{n-1} + t_{n-2}
\end{aligned}$$

Of course, for each one of these recurrences, we have appropriate termination conditions. Before we solve the equations, we will state a theorem without proof. The results of the theorem can be used to solve homogeneous recurrence relations with constant coefficients.

Theorem 4.1 *Consider the homogeneous linear recurrence relations with constant coefficients:*

$$a_0\, t_n + a_1\, t_{n-1} + \cdots a_k\, t_{n-k} = 0. \tag{4.4}$$

Its characteristic equation is obtained by assuming $t_n = r^n$ *for an arbitrary* r *and by making substitutions in the equation giving*

$$a_o\, r^k + a_1\, r^{k-1} + \cdots + a_k\, r^0 = 0 \tag{4.5}$$

after algebraic simplification. This is the characteristic equation and if it has k *distinct solutions* $r_1, r_2, \cdots, r_k$*, then the only solutions to the recurrence are*

$$t_n = c_1\, r_1^n + c_2\, r_2^n + \cdots + c_k\, r_k^n \tag{4.6}$$

where the c_i *terms are arbitrary constants.*

Using the results of this theorem, we go through the following steps to obtain the solutions to a homogeneous linear recurrence with constant coefficients. The steps are similar to those used in solving homogeneous linear differential equations. Recurrence relations are not differential equations, but they are *difference equations*. Difference equations relate various terms in an arbitrary sequence.

1. Rewrite the recurrence by moving all t_i terms to the right hand side and equating the left hand side to 0.

2. Assume $t_n = r^n$ is a solution to the recurrence relation for an arbitrary r. Substitute r^n for t_n in the recurrence relation and simplify to obtain the characteristic equation.

3. Solve the characteristic equation to obtain its roots. For the time being, assume that the roots are not multiple roots. A multiple root is one that is found to be a root several times.

4. Write the solution to the original recurrence as

$$t_n = c_1\, r_1^n + c_2\, r_2^n + \cdots + c_k\, r_k^n$$

   where each c_i is an arbitrary constant.

5. Obtain the values of the k arbitrary constants c_i by considering the initial conditions for the recurrence relation. There must be k such initial conditions to obtain the values of the k constants.

We solve a few homogeneous linear recurrence relations with constant coefficients below.

4.1.1 Solving the Recurrence Relation $t_n = 4\, t_{n-1}$

Consider the recurrence relation

$$\begin{aligned} t_n &= 4\, t_{n-1} && n \geq 2 \\ t_1 &= 1 \end{aligned} \tag{4.7}$$

This is a very simple recurrence relation and can be easily solved by Repeated Substitution following the discussions in Chapter 3. However, we solve it using characteristic equations to demonstrate how the technique works.

We go through the general steps outlined in the previous section.

Step 1: Rewrite the original recurrence as

$$t_n - 4\, t_{n-1} = 0.$$

Step 2: We assume that $t_n = r^n$ is a solution to our recurrence relation for some arbitrary r. We substitute r^n for t_n in the recurrence to obtain

$$\begin{aligned} r^n - 4\, r^{n-1} &= 0 \\ r^{n-1}(r-4) &= 0. \end{aligned}$$

So the characteristic equation is

$$r - 4 = 0.$$

We ignore the r^{n-1} part since it leads to the trivial root $r = 0$.

Step 3: We solve the characteristic equation to obtain $r_1 = 4$ as the only root. It is a not a multiple root. Thus, the only solution to the characteristic equation is $r_1 = 4$.

Step 4: We write the solution to the original recurrence as

$$\begin{aligned} t_n &= c_1\, r_1^n \\ &= c_1\, 4^n. \end{aligned}$$

for some constant c_1.

Step 5: We have only one constant to evaluate. We have an initial condition $t_1 = 1$. This gives us the following.

$$\begin{aligned} t_1 &= 1 \\ c_1\, 4 &= 1 \\ c_1 &= \frac{1}{4} \end{aligned}$$

Therefore, the solution to the original recurrence is

$$\begin{aligned} t_n &= \frac{1}{4}\, 4^n \\ &= 4^{n-1}. \end{aligned}$$

Thus, the solution to the recurrence relation 4.7 is

$$t_n = 4^{n-1}. \tag{4.8}$$

4.1.2 Solving the Recurrence Relation $t_n = t_{n-1} + t_{n-2}$

The recurrence we want to solve is given below.

$$\begin{aligned} t_n &= t_{n-1} + t_{n-2} && n \geq 2 \\ t_0 &= 1 \\ t_1 &= 1 \end{aligned} \tag{4.9}$$

The recurrence above is a homogeneous recurrence relation with constant coefficients. This recurrence defines what are called Fibonacci numbers. The solution to this recurrence relations follows.

Step 1: Rewrite the recurrence with all the t_i terms on the left hand side.

$$t_n - t_{n-1} - t_{n-2} = 0 \qquad n \geq 2$$

Step 2: We assume the solution to the recurrence relation is $t_n = r^n$ for some arbitrary r. We substitute r^n for t_n to obtain

$$\begin{aligned} r^n - r^{n-1} - r^{n-2} &= 0 \\ r^{n-2}(r^2 - r - 1) &= 0 \\ r^2 - r - 1 &= 0. \end{aligned}$$

This is the characteristic equation. We ignore the trivial solution $r = 0$.

Step 3: We solve the characteristic equation to obtain its roots. We get

$$\begin{aligned} r &= \frac{1 \pm \sqrt{1+4}}{2} \\ &= \frac{1 \pm \sqrt{5}}{2}. \end{aligned}$$

Let us call the two roots r_1 and r_2. Let

$$r_1 = \frac{1 + \sqrt{5}}{2}$$

and

$$r_2 = \frac{1 - \sqrt{5}}{2}.$$

Step 4: We write the solution to the original recurrence as

$$t_n = c_1\, r_1^n + c_2\, r_2^n \tag{4.10}$$

for two arbitrary constants c_1 and c_2. The values of c_1 and c_2 can be obtained by using the two termination conditions of the original recurrence: $t_0 = 1$ and $t_1 = 1$. $t_0 = 1$ gives us

$$1 = c_1 + c_2 \tag{4.11}$$

and $t_1 = 1$ gives us

$$1 = c_1 \left(\frac{1+\sqrt{5}}{2}\right) + c_2 \left(\frac{1-\sqrt{5}}{2}\right). \tag{4.12}$$

We can solve for c_1 and c_2 using Equations 4.11 and 4.12 to obtain

$$\begin{aligned} c_1 &= \frac{\sqrt{5}-1}{2\sqrt{5}}, \text{and} \\ c_2 &= \frac{\sqrt{5}+1}{2\sqrt{5}}. \end{aligned}$$

Thus, the solution to the recurrence relation 4.9 is obtained.

$$\begin{aligned} t_n &= c_1\, r_1^n + c_2\, r_2^n \\ &= \frac{\sqrt{5}+1}{2\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{\sqrt{5}-1}{2\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n \\ &= \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^{n+1} \\ &= \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right] \end{aligned} \tag{4.13}$$

If we call $\alpha_1 = \frac{1+\sqrt{5}}{2}$ and $\alpha_2 = \frac{1-\sqrt{5}}{2}$, then

$$t_n = \frac{1}{\sqrt{5}} \left[\alpha_1^{n+1} - \alpha_2^{n+1}\right]. \tag{4.14}$$

Note that the value of $\alpha_1 \approx 1.619$ and $\alpha_2 \approx -0.619$. Since $\mid \alpha_2 \mid < 1$, if n is large, Equation 4.14 can be written as

$$t_n = \frac{1}{\sqrt{5}} \alpha_1^{n+1}. \tag{4.15}$$

4.2 Homogeneous Recurrences with Multiple Roots

It is possible that when we solve for the roots of a characteristic equation, one or more of the roots are multiple roots. A multiple root occurs several times. For example, let the characteristic equation be $r^2 - 2r + 1 = 0$. This equation can be rewritten as $(r-1)^2 = 0$. Thus, the two roots are both 1. We say that this equation has a root of multiplicity 2. When a characteristic equation for a recurrence relation has a multiple root, we slightly change the method outlined in the previous section. This requires the statement of another theorem.

Theorem 4.2 *Consider the homogeneous linear recurrence relations with constant coefficients:*

$$a_0\, t_n + a_1\, t_{n-1} + \cdots + a_k\, t_{n-k} = 0. \tag{4.16}$$

We assume $t_n = r^n$ for an arbitrary r and make substitutions in the equation giving

$$a_o\, r^k + a_1\, r^{k-1} + \cdots + a_k\, r^0 = 0 \tag{4.17}$$

as the characteristic equation. Assume that the characteristic equation has s roots $r_1, r_2, \cdots, r_s$ with multiplicities $m_1, m_2, \cdots, m_s$, respectively such that $\sum_1^s\ m_i = k$. Then the solutions to the linear recurrence can be written as

$$t_n = \sum_{i=1}^{s} t_{i,n} \tag{4.18}$$

where

$$t_{i,n} = \left(c_{i,0} + c_{i,1}\, n + c_{i,2}\, n^2 \cdots + c_{i,m_{i-1}}\, n^{m_i - 1}\right) \tag{4.19}$$

where the $c_{i,j}$ terms are arbitrary constants.

Obviously, the theorem can be used to solve a linear recurrence whose characteristic equation has one or several multiple roots. Here, the assumption is that the ith root is a multiple root of degree m_{i-1}. When the ith root is of degree one, it contributes only one term to the solution of the recurrence relation, as we saw in Section 4.1. However, in the theorem, the ith root is a multiple root of multiplicity m_i. So, it contributes m_i terms to the solution of the recurrence relation. The terms it contributes are given by the theorem. If we assume that each root is a multiple root (of possibly, degree 1 if it is a single root), then we can write the complete solution to the recurrence relation as a sum of the contributions of each multiple root in light of the the theorem above. This is written in the theorem as $t_n = \sum_{i=1}^{s} t_{i,n}$.

We rewrite below the steps needed to solve a homogeneous linear equation where the characteristic equation has multiple roots. It is a variation of the steps we have already seen in Section 4.1.

1. Rewrite the recurrence by moving all t_i terms to the right hand side and equating the left hand side to 0.

2. Assume $t_n = r^n$ is a solution to the recurrence relation for an arbitrary r. Substitute r^n for t_n in the recurrence relation and simplify to obtain the characteristic equation.

3. Solve the characteristic equation to obtain its roots. Assume that the characteristic equation has s roots $r_1, r_2, \cdots, r_s$ with multiplicities $m_1, m_2, \cdots, m_s$, respectively such that $\sum_1^s\ m_i = k$.

4. Write the solution to the original recurrence as

   $$t_n = \sum_{i=1}^{s} t_{i,n}$$

   where

   $$t_{i,n} = \left(c_{i,0} + c_{i,1}\, n + c_{i,2}\, n^2 \cdots + c_{i,m_i - 1}\, n^{m_i - 1}\right)$$

   where the $c_{i,j}$ terms are arbitrary constants.

5. Obtain the values of the k arbitrary constants $c_{i,j}$ by considering the k initial conditions for the recurrence relation. There must be k such initial conditions to obtain the values of the k constants.

We solve a couple of linear homogeneous recurrence relations whose characteristic equations have multiple roots, using the steps outlined above.

4.2.1 Solving the Recurrence $t_n - 7\, t_{n-1} + 15\, t_{n-2} - 9\, t_{n-3} = 0$

The recurrence we want to solve is

$$\begin{aligned} t_n - 7\, t_{n-1} + 15\, t_{n-2} - 9\, t_{n-3} &= 0 \qquad n > 2 \\ t_0 &= 0 \\ t_1 &= 1 \\ t_2 &= 2. \end{aligned} \tag{4.20}$$

We have the recurrence relation and three initial or, termination conditions.

To solve, we go through the steps outlined above.

Step 1: The recurrence is already given in a way that all the t_i terms are on the left hand side. So, nothing needs to be done in this step.

Step 2: We obtain the characteristic equation by assuming that $t_n = r^n$. This gives us the characteristic equation as

$$r^3 - 7\, r^2 + 15\, r - 9 = 0.$$

Step 3: We solve the characteristic equation.

$$\begin{aligned} r^3 - 7\, r^2 + 15\, r - 9 &= 0 \\ r^3 - r^2 - 6\, r^2 + 6\, r + 9\, r - 9 &= 0 \\ r^2(r-1) - 6\, r(r-1) + 9(r-1) &= 0 \\ (r-1)(r^2 - 6\, r + 9) &= 0 \\ (r-1)(r-3)^2 &= 0 \end{aligned}$$

Let the roots be r_1 and r_2. We can arbitrarily let $r_1 = 1$, a root of multiplicity 1; and $r_2 = 3$ be a root of mulitiplicity 2.

Step 4: We write the solution to the original recurrence as

$$\begin{aligned} t_n &= c_1\, r_1^n + (c_{2,1} + c_{2,2}\, n)\ r_2^n \\ &= c_1\, 1^n + (c_{2,1} + c_{2,2}\, n)\ 3^n \\ &= c_1 + c_{2,1}\, 3^n + c_{2,2}\, n\, 3^n \end{aligned}$$

Step 5: We determine the constants from the initial conditions. The three initial conditions $t_0 = 0$, $t_1 = 1$ and $t_2 = 2$ respectively give us the following equations:

$$c_1 + c_{2,1} = 0$$

$$\begin{aligned} c_1 + 3\ c_{2,1} + 3\ c_{2,2} &= 1, \text{and} \\ c_1 + 9\ c_{2,1} + 18\ c_{2,2} &= 2. \end{aligned}$$

This is a linear system of equations that we solve. The solution gives us $c_1 = -1$, $c_{2,1} = 1$ and $c_{2,2} = -\frac{1}{3}$. With these values, the solution to the recurrence relation becomes

$$\begin{aligned} t_n &= -1 + 3^n - \frac{1}{3}\ n\ 3^n \\ &= -1 + 3^n - n\ 3^{n-1}. \end{aligned} \tag{4.21}$$

4.2.2 Solving the Recurrence $t_n - 5\ t_{n-1} + 7\ t_{n-2} - 3\ t_{n-3} = 0$

The recurrence we want to solve is

$$\begin{aligned} t_n - 5\ t_{n-1} + 7\ t_{n-2} - 3\ t_{n-3} &= 0 \qquad\qquad n > 2 \\ t_0 &= 1 \\ t_1 &= 2, \text{and} \\ t_2 &= 3. \end{aligned} \tag{4.22}$$

To solve, we go through the steps outlined above.

Step 1: The recurrence is already given in a way that all the t_i terms are on the left hand side. So, nothing needs to be done in this step.

Step 2: We obtain the characteristic equation by assuming that $t_n = r^n$. This gives us the characteristic equation as

$$r^3 - 5\ r^2 + 7\ r - 3 = 0.$$

Step 3: We solve the characteristic equation.

$$\begin{aligned} r^3 - 5\ r^2 + 7\ r - 3 &= 0 \\ r^3 - r^2 - 4\ r^2 + 4\ r + 3\ r - 3 &= 0 \\ r^2(r-1) - 4\ r(r-1) + 3(r-1) &= 0 \\ (r-1)(r^2 - 4\ r + 3) &= 0 \\ (r-1)(r^2 - 3\ r - r + 3) &= 0 \\ (r-1)(r[r-3] - [r-3]) &= 0 \\ (r-1)(r-1)(r-3) &= 0 \\ (r-1)^2\ (r-3) &= 0 \end{aligned}$$

Let the roots be r_1 and r_2. We can arbitrarily let $r_1 = 1$, a root of multiplicity 2; and $r_2 = 3$ be a root of mulitiplicity 1.

Step 4: We write the solution to the original recurrence as

$$\begin{aligned} t_n &= (c_{1,1} + c_{1,2}\ n)\ r_1^n + c_2\ r_2^n \\ &= (c_{1,1} + c_{1,2}\ n)\ 1^n + c_2\ 3^n \\ &= c_{1,1} + c_{1,2}\ n + c_2\ 3^n \end{aligned}$$

Step 5: We next determine the constants from the initial conditions. The three initial conditions $t_0 = 1$, $t_1 = 2$ and $t_2 = 3$ respectively give us the following equations:

$$\begin{aligned} c_{1,1} + c_2 &= 1 \\ c_{1,1} + c_{1,2} + 3\, c_2 &= 2 \\ c_{1,1} + 2\, c_{1,2} + 9\, c_2 &= 3. \end{aligned}$$

This is a linear system of equations that we need to solve. The solution gives us $c_{1,1} = 1$, $c_{1,2} = 1$ and $c_2 = 0$. With these values, the solution to the recurrence relation becomes

$$\begin{aligned} t_n &= 1 + n + 0 \times\ 3^n \\ &= n + 1. \end{aligned} \tag{4.23}$$

4.3 Non-homogeneous Linear Recurrences

If $f(n) \neq 0$, a recurrence of the form

$$a_0\, t_n + a_1\, t_{n-1} + \cdots + a_k\, t_{n-k} = f(n) \tag{4.24}$$

where k and a_i terms are constants is called a *non-homogeneous linear recurrence equation with constant coefficients*. There is no general method for solving a non-homogeneous linear recurrence equation.

There is a special case that is fairly common that can be solved easily. The special case is

$$a_0\, t_n + a_1\, t_{n-1} + \cdots + a_k\, t_{n-k} = b^n\, p(n) \tag{4.25}$$

where b is a constant and $p(n)$ is a polynomial in n. Such a recurrence relation can be solved by converting to a homogeneous recurrence relation. We solve a couple of non-homogeneous linear recurrences below.

4.3.1 Solving the Recurrence $t_n - 3\, t_{n-1} = 4^n$

The recurrence we want to solve is

$$\begin{aligned} t_n - 3\, t_{n-1} &= 4^n \\ t_0 &= 0 \\ t_1 &= 4. \end{aligned} \tag{4.26}$$

The first is the actual recurrence, the other two are initial or termination conditions. This is not a homogeneous recurrence because of the term 4^n on the right hand side in the recurrence. We can easily get rid of this term by algebraic manipulation.

To remove 4^n from the right hand side, we need to instantiate the recurrence Equation 4.45 for $n-1$. In other words, we substitute $n-1$ for n everywhere in the equation to obtain

$$t_{n-1} - 3\, t_{n-2} = 4^{n-1}.$$

We multiply this equation by 4 all throughout to obtain

$$4\, t_{n-1} - 12\, t_{n-2} = 4^n. \tag{4.27}$$

If we subtract 4.27 from 4.26, we get the following.

$$\begin{array}{ccccccc} t_n & - & 3\, t_{n-1} & & & = & 4^n \\ & & 4\, t_{n-1} & - & 12\, t_{n-2} & = & 4^n \\ \hline t_n & - & 7\, t_{n-1} & + & 12\, t_{n-2} & = & 0 \end{array} \tag{4.28}$$

Equation 4.28 obtained by subtracting is a homogeneous linear recurrence with constant coefficients. We can solve it using the characteristic equation method.
The characteristic equation for the recurrence relation is

$$r^2 - 7\, r + 12 = 0$$

This characteristic equation has two roots, 3 and 4. Let us say $r_1 = 3$ and $r_2 = 4$ are the roots. Each root is of multiplicity 1. Therefore, the solution the recurrence is

$$\begin{aligned} t_n &= c_1\, r_1^n + c_2\, r_2^n \\ &= 3^n\, c_1 + 4^n\, c_2 \end{aligned} \tag{4.29}$$

We obtain the values of the two constants c_1 and c_2 from the initial values, viz., $t_0 = 0$ and $t_1 = 1$. By using these initial conditions, we get the following two equations.

$$c_1 + c_2 = 0 \tag{4.30}$$
$$3\, c_1 + 4\, c_2 = 4 \tag{4.31}$$

Solving these two equations, we get $c_1 = -4$ and $c_2 = 4$. Therefore, the solution to the recurrence equation becomes

$$\begin{aligned} t_n &= -4 \times 3^n + 4 \times 4^n \\ &= 4^{n+1} - 4 \times 3^n \end{aligned} \tag{4.32}$$

4.3.2 Solving the Recurrence $t_n - t_{n-1} = cn$

This is the recurrence we have solved before in the previous chapter. It was earlier written as

$$T(n) = T(n-1) + cn$$

We can covert this to the notation used in the current chapter by writing it as

$$\begin{aligned} t_n &= t_{n-1} + cn, \text{or} \\ t_n - t_{n-1} &= cn. \end{aligned} \tag{4.33}$$

after we move the t_i terms to the left of the equality sign. This is a non-homogeneous recurrence. We need to convert this recurrence to a homogeneous recurrence relation. We do that by instantiating the recurrence for $n-1$ and subtracting the new equation from the original recurrence:

$$\begin{array}{ccccccc} t_n & - & t_{n-1} & & & = & c(n-1) \\ & & t_{n-1} & - & t_{n-2} & = & c(n-1) \\ \hline t_n & - & 2\,t_{n-1} & + & t_{n-2} & = & c. \end{array} \tag{4.34}$$

This is not a homogeneous linear recurrence. We have reduced the right hand side of the recurrence from a linear function in n to a constant function c. We need to go through the process of substitution and subtraction one more time to obtain the linear homogeneous equation as

$$t_n - 3\,t_{n-1} + 3\,t_{n-2} - t_{n-3} = 0. \tag{4.35}$$

The characteristic equation is

$$r^3 - 3\,r^2 + 3\,r - 1 = 0. \tag{4.36}$$

We can easily see that it can be written as

$$(r-1)^3 = 0.$$

Therefore the characteristic equation has only one root $r_1 = 1$ of mulitiplicity 3. This gives the solution to the recurrence relation as

$$\begin{aligned} t_n &= \left(c_{1,1} + c_{1,2}\,n + c_{1,3}\,n^2\right) r_1^n \\ &= c_{1,1} + c_{1,2}\,n + c_{1,3}\,n^2. \end{aligned} \tag{4.37}$$

$$\tag{4.38}$$

There are three constants to be determined and we can obtain their values from three initial conditions.

When we solved this recurrence in the previous chapter, we needed only one initial condition, viz, $t(1) = d$ or $t_n = d$ in the notation used in this section. However, we need three initial conditions now, not just one. Assume the initial conditions given are $t_1 = 1$, $t_2 = 7$ and $t_3 = 13$. Based on these initial conditions, we get

$$\begin{aligned} c_{1,1} + c_{1,2} + c_{1,3} &= 3 \\ c_{1,1} + 2\,c_{1,2} + 4\,c_{1,3} &= 3 \\ c_{1,1} + 3\,c_{1,2} + 9\,c_{1,3} &= 3 \end{aligned} \tag{4.39}$$

respectively. The solution of this set of linear equations is $c_{1,1} = 1, c_{1,2} = 1$ and $c_{1,3} = 1$. As a result, the solution to the recurrence is

$$t_n = n^2 + n + 1. \tag{4.40}$$

4.3.3 Generalizing the Solution

We see that a non-homogeneous linear recurrence relation with constant coefficients can be converted to a homogeneous linear recurrence fairly easily under the condition that the the right hand side is of the form $b^n\ p(n)$ where b is a constant and $p(n)$ is a polynomial of degree d. But, it may take one, two, three or more successive iterations to reduce the degree of the polynomial on the right hand side. In each iteration, the degree of the polynomial comes down by one. In each iteration, we add a factor to the characteristic equation of the form $(r - b)$ where r is the root we are seeking and b is the constant that appears on the right hand side of the recurrence relation. So, if we have $b^n\ p(n)$ on the right hand side, in each iteration in our attempt to homogenize the recurrence, we reduce the degree of the polynomial on the right hand side by 1. It takes d iterations to reduce the polynomial to constant, and one more iteration to reduce to the right hand side to 0. As a result, $(r - b)^{(d+1)}$ appears in the characteristic equation of the homogenized linear recurrence. This leads to general way to solve such recurrence relations.

Theorem 4.3 *Suppose we have a recurrence relation of the form*

$$a_0\ t_n + a_1\ t_{n-1} + \cdots a_k\ t_k = b^n\ p(n) \tag{4.41}$$

where b is a constant and $p(n)$ is a polynomial in n of degree n. The characteristic equation for the polynomial can be written as a product of two parts: a part contributed by the corresponding homogeneous recurrence obtained by setting the right hand side to 0, and a part obtained from the non-homogenous part or the non-zero right hand side $b^n\ p(n)$. The contribution to the characteristic equation from the homogeneous recurrence is

$$(a_o\, r^k + a_1\ r^{k-1} + \cdots + a_k) \tag{4.42}$$

and the contribution of the non-zero right hand side is

$$(r - b)^{d+1}. \tag{4.43}$$

Therefore, the final characteristic equation is

$$(a_o\, r^k + a_1\ r^{k-1} + \cdots + a_k) \times (r - b)^{d+1}. \tag{4.44}$$

4.3.4 Resolving the Recurrence $t_n - 3\, t_{n-1} = 4^n$

The recurrence we want to solve using the theorem above is

$$\begin{aligned} t_n - 3\, t_{n-1} &= 4^n \\ t_0 &= 0 \\ t_1 &= 4. \end{aligned} \quad (4.45)$$

The characteristic equation has two parts. One part coming from the homogeneous recurrence relation is $(r-3)$. The part coming from the non-homogeneous part is $(r-b)^{d+1}$. Here $b = 4$ and $d = 0$ since the polynomial $p(n)$ is actually the constant polynomial 1. Therefore, the contribution of the non-homogeneous part is $(r-4)^{0+1}$ or $(r-4)$.
The final characteristic equation for the recurrence relation is

$$(r-3)(r-4) = 0.$$

This characteristic equation has two roots, 3 and 4. Let us say $r_1 = 3$ and $r_2 = 4$ are the roots. Each root is of multiplicity 1. Therefore, the solution the recurrence is

$$\begin{aligned} t_n &= c_1\, r_1^n + c_2\, r_2^n \\ &= 3^n\, c_1 + 4^n\, c_2 \end{aligned} \quad (4.46)$$

As before, we can obtain the two constants using the initial conditions.

4.3.5 Resolving the Recurrence $t_n - t_{n-1} = cn$

This is the recurrence we have solved before in the previous chapter as well as in a previous section in this chapter. This is not a homogeneous linear recurrence. The right hand side is of the form $b^n\, p(n)$ where $b = 1$ and $p(n) = cn$. The characteristic equation is composed of two parts. The part due to the corresponding homogeneous recurrence relation is

$$(r-1) \quad (4.47)$$

The contribution to the characteristic equation from the non-homogeneous part is $(r-b)^{d+1}$ where d is the degree of the polynomial $p(n)$. Here, $b = 1$ and $d = 1$. Therefore, the contribution from the non-homogeneous part is $(r-b)^2$. Therefore, the complete recurrence relation is

$$(r-1)^3 = 0. \quad (4.48)$$

Therefore the characteristic equation has only one root $r_1 = 1$ of mulitiplicity 3. This gives the solution to the recurrence relation as

$$\begin{aligned} t_n &= \left(c_{1,1} + c_{1,2}\, n + c_{1,3}\, n^2\right) r_1^n \\ &= c_{1,1} + c_{1,2}\, n + c_{1,3}\, n^2 \end{aligned} \quad (4.49)$$

$$(4.50)$$

There are three constants and we can obtain their values from three initial conditions.

Chapter 5

Generating Functions

A recurrence relation for a function $T()$ is given in terms of an equation that contains $T()$ on the left hand side as well as on the right hand side. The argument of $T()$ on the right hand side is smaller than the argument of $T()$ on the left hand side. In addition, one or more terminating conditions are given for $T()$. When we solve a recurrence relation, we get a general expression for $T(n)$. Along with the termination conditions, the general formula allows us to obtain values of the function $T(n)$ for various integer values of n, starting usually from either 0 or 1. Thus, we can obtain the sequence of values:

$$T(0), T(1), T(2), \cdots, T(n), \cdots$$

for successive values of n.

These successive values of $T(n)$ can be used to write an infinite power series $g(x)$.

$$g(x) = T(0) + T(1)\ x + T(2)\ x^2 + \cdots + T(n)\ x^n + \cdots \tag{5.1}$$

Here, x is a dummy variable. $T(i), i \geq 0$ is the coefficient of x^i. Successive terms of $g(x)$ are values of the function $T(n)$ for increasing integer values of n. Such a function $g(x)$ is called a generating function for $T(n)$. That is because values of the function $T()$ defined by a recurrence relation can be generated if we know the coefficients of powers of x in $g(x)$.

Thus, a generating function is an infinite power series. A power series is obtained by summing terms where the powers of a variable x increases successively, starting from the zeroth power. Two types of generating functions are generally used to solve recurrence relations. They are,

- Ordinary generating functions, and
- Exponential generating functions.

The generating function $g(x)$ we see above is an ordinary generating function. In this Chapter, we look at how ordinary generating functions can be used to solve recurrence relations.

5.1 Preliminaries

A few infinite power series expansions for fractions are very commonly used in solving problems with recurrence relations. For example, quite frequently we need to know the infinite series expansion for $\frac{1}{1-x}$, $\frac{1}{(1-x)^2}$, $\frac{1}{(1-x)^3}$, etc. In the following, we obtain infinite series expansions for these fractions.

5.1.1 Power Series Expansion for $\frac{1}{1-x}$

The following is the expansion for $\frac{1}{1-x}$.

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots + x^n + \cdots \tag{5.2}$$

Since, a power series can be characterized as a summation of its general term, we can write the following.

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n \tag{5.3}$$

This equation can be obtained in many ways.

1. One can simply divide in long hand and obtain the expansion.
2. Another way to obtain the expansion is to sum the infinite series on the right hand side.
3. A third way is to produce a Taylor's series expansion.

Obtaining the expansion by long division is straight-forward. The second approach to obtain the expansion is to sum the geometric series on the right hand side. It is a geometric series with x as the constant ratio between two consecutive terms. The sum of the geometric series given above is the following.

$$1 + x + x^2 + \cdots + x^n = \frac{1-x^n}{1-x} \tag{5.4}$$

Now, if $x < 1$ and $n \to \infty$, we can write the following.

$$1 + x + x^2 + \cdots = \frac{1}{1-x} \tag{5.5}$$

The third way to obtain the value is by using Taylor's series and the Binomial Theorem. We discuss this technique a little later.

5.1.2 Power Series Expansion for $\frac{1}{(1-x)^i}, i > 1$

Often, we need to obtain the power series expansion for fractions such as $\frac{1}{(1-x)^2}$ or $\frac{1}{(1-x)^3}$. These can be derived from the power series expansion of $\frac{1}{1-x}$. From the expression for $\frac{1}{1-x}$, we can get an expression for $\frac{1}{(1-x)^2}$ by differentiating both sides.

$$\begin{aligned} -1(1-x)^{-2}(-1) &= 1 + 2x + 3x^2 + \cdots + nx^{n-1} + (n+1)x^n + \cdots \\ \frac{1}{(1-x)^2} &= 1 + 2x + 3x^2 + \cdots + nx^{n-1} + (n+1)x^n + \cdots \end{aligned} \tag{5.6}$$

Note that the coefficient of x^n in the power series expansion of $\frac{1}{(1-x)^2}$ is $n+1$. This is an important fact that finds use in solving problems with generating functions. We can also write

$$\frac{1}{(1-x)^2} = \sum_{n=0}^{\infty} (n+1)\ x^n. \tag{5.7}$$

Differentiating one more time, we get the following.

$$\begin{aligned} -2(1-x)^{-3}(-1) &= 2 + 2 \times 3x + \cdots + n(n+1)x^{n-1} + n(n+1)x^n + \cdots \\ \frac{2}{(1-x)^3} &= 2 + 2 \times 3x + \cdots + n(n+1)x^{n-1} + n(n+1)x^n + \cdots \end{aligned}$$

From this expansion, we know that the coefficient of x^n in the power series expansion of $\frac{1}{(1-x)^3}$ is $\frac{1}{2}(n+1)(n+2)$. Therefore, we can write

$$\frac{1}{(1-x)^3} = \frac{1}{2} \sum_{n=0}^{\infty} (n+1)(n+2)\ x^n \tag{5.8}$$

5.1.3 Power Series Expansion of $\frac{1}{(1-ax)^i}, i \geq 1$

Another expression that is commonly used the following.

$$\begin{aligned} \frac{1}{1-ax} &= 1 + ax + (ax)^2 + \cdots + (ax)^n + \cdots \\ &= 1 + ax + a^2x^2 + \cdots + a^nx^n \end{aligned} \tag{5.9}$$

Once again, the expression for $\frac{1}{1-ax}, 1 - ax \neq 0$, can be obtained in several ways, just like we obtained the value of $\frac{1}{1-x}$. The coefficient of x^n in the power series expansion of $\frac{1}{1-ax}$ is a^n.

Differentiating once, we get the following.

$$-1(1-ax)^{-2}(-1) = \frac{a}{(1-ax)^2} = a + 2a^2x + 3a^3x^2 + \cdots + na^nx^{n-1} + (n+1)a^{n+1}x^n + \cdots \tag{5.10}$$

The coefficient of x^n in the power series expansion of $\frac{1}{(1-ax)^2}$ is $(n+1)a^n$.

5.1.4 Power Series Expansion Using the Binomial Theorem

In the previous sections, we have showed how we can obtain the power series expansions for $\frac{1}{1-x}$, $\frac{1}{(1-x)^2}$, $\frac{1}{(1-x)^3}$, etc. These fractions can also be written as $(1-x)^{(-1)}$, $(1-x)^{(-2)}$, $(1-x)^{(-3)}$, etc. Thus, each fraction can be considered as raising $(1-x)$ to a certain negative integer power.

We can use the Binomial Theorem to obtain the value of $(1+x)^i, i \geq 1$, when i is in an integer. So, the Binomial Theorem can be used to obtain expansions for $(1+x)^2, (1+x)^{20}, (1+x)^{100}$, etc. We usually learn how to obtain such expansions in high school.

The Binomial Theorem, or an extension of it, called the *Generalized Binomial Theorem*, can be used to obtain values of $(1+x)^i$ for any rational real value of i. In other words, we can use the Generalized Binomial Theorem to obtain an expansion for $(1+x)^i$, even if i is a negative integer, a fraction, or any other real number. Therefore, the Generalized Binomial Theorem can be used to obtain expansions for $(1+x)^{-1}, (1-x)^{-2}, (1-x)^{\sqrt{5}}, \sqrt{1+x}$, etc.

The Generalized Binomial Theorem can be derived from Taylor's Series. If we know the value of a function $f(x)$ at a point a and the derivatives of the function at this point a, the value of $f(x)$ is given as follows.

$$f(a+x) = f(a) + x\, f'(a) + x^2\, \frac{f''(a)}{2!} + \cdots + x^k\, \frac{f^{(k)}(a)}{k!} + \cdots \tag{5.11}$$

$f(a)$ is the value of the function at the point a. $f'(a), f''(a), \cdots, f^{(n)}(a)$, etc., are successive derivatives of the function $f(x)$ at the point a.

We can use Taylor's series expansion in the neighborhood of any point a. A convenient value of a is 1. If $a = 1$, we get

$$f(1+x) = f(1) + xf'(1) + \frac{x^2}{2!}f''(1) + \cdots + \frac{x^k}{k!}f^{(k)}(1) + \cdots + \tag{5.12}$$

This formula works for any function $f(x)$ that is continuous at $x = 1$ and has continuous derivatives at $x = 1$. In particular, we can let $f(x) = x^n$ for any n. The following table gives the values of $f(x)$ and its derivatives at the point $x = 1$.

General Values	Values at $x = 1$
$f(x) = x^n$	$f(1) = 1$
$f'(x) = n\, x^{n-1}$	$f'(1) = n$
$f''(x) = n(n-1)\, x^{n-2}$	$f''(1) = n(n-1)$
$\vdots$	$\vdots$
$f^{(k)}(x) = n(n-1)\cdots(n-k+1)\, x^{n-k}$	$f^{(k)}(1) = n(n-1)\cdots(n-k+1)$

Therefore, in general, the expansion for $(1+x)^n$ can be written as follows.

$$(1+x)^n = 1 + nx + \frac{1}{2!}n(n-1)x^2 + \cdots + \frac{1}{k!}\left[n(n-1)\cdots(n-k+1)\,\right]\, x^k \tag{5.13}$$

This formula can be used for any real value of n. This is the Generalized Binomial Theorem. If n is a positive integer, we can perform some simplifications to the expression for $(1+x)^n$. The value of $f^{(k)}(1)$ can be expressed in terms of factorials if we multiply the expression and divide it by $(n-k)!$. k is assumed always to be a positive integer.

$$\begin{aligned} f^{(k)}(1) &= n(n-1)\cdots(n-k+1) \\ &= \frac{n(n-1)\cdots(n-k+1)(n-k)!}{(n-k)!} \\ &= \frac{n!}{(n-k)!} \end{aligned}$$

Thus, for a positive integer n, we can write the following.

$$\begin{aligned} (1+x)^n &= 1 + \frac{n!}{1!(n-1)!}x + \frac{n!}{2!(n-2)!}x^2 + \cdots + \frac{n!}{(n-k)!k!}x^k + \cdots + \frac{n!}{n!0!}x^n \\ &= \sum_{k=0}^{n} \frac{n!}{(n-k)!k!}x^k \\ &= \sum_{k=0}^{n} \binom{n}{k} x^k \end{aligned} \tag{5.14}$$

Because coefficients after x^n all become 0, the infinite series becomes a finite series if n is a positive integer.

Problem 14 *Find the expansion for $(1+x)^5$. Also, find the coefficient of x^5 in $(1+x)^5$.*

Here, $n=5$. Therefore, the terminating Binomial Theorem can be used.

$$\begin{aligned} (1+x)^5 &= \frac{5!}{5!\,0!}x^0 + \frac{5!}{4!\,1!}x^1 + \frac{5!}{3!\,2!}x^2 + \frac{5!}{2!\,3!}x^3 + \frac{5!}{1!\,4!}x^4 + \frac{5!}{0!\,5!}x^5 \\ &= 1 + 5x + 10x^2 + 10x^3 + 5x^4 + x^5 \end{aligned}$$

The coefficient of x^5 in the expression for $(1+x)^5$ is 1.

Problem 15 *Find the coefficient of x^k in the expansion of $\frac{1}{1+x} = (1+x)^{-1}$. Obtain the expansion for $\frac{1}{1+x}$. Assume k is a non-negative integer.*

The coefficient of $x^k, k \geq 0$, in the expansion of $(1+x)^n$ is given as

$$\frac{1}{k!}n(n-1)\cdots(n-k+1)$$

$n=-1$. Therefore, the coefficient of $x^k, k \geq 0$ is

$$\frac{1}{k!}(-1)(-1-1)\cdots(-1-k+1)$$

$$
\begin{aligned}
&= \frac{1}{k!}(-1)(-2)(-3)\cdots(-k) \\
&= \frac{1}{k!}(-1)^k k! \\
&= (-1)^k
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\frac{1}{1+x} &= (-1)^0x^0 + (-1)^1x^1 + (-1)^2x^2 + \cdots + (-1)^kx^k + \cdots \\
&= 1 - x + x^2 - x^3 + \cdots + (-1)^k\, x^k + \cdots
\end{aligned}
$$

Problem 16 *Find the coefficient of x^k in the expansion of $\frac{1}{(1-x)^2} = (1-x)^{-2}$. Obtain the series expansion for $\frac{1}{(1-x)^2}$.*

The coefficient of $x^k, k \geq 0$, in the expansion of $(1+x)^n$ is

$$\frac{1}{k!}n(n-1)\cdots(n-k+1)$$

Here $n = -2$. Therefore, the coefficient of $x^k, k \geq 0$, is

$$
\begin{aligned}
&\frac{1}{k!}(-2)(-2-1)\cdots(-2-k+1) \\
&= \frac{1}{k!}(-1)^k 2 \times 3 \cdots \times (k+1) \\
&= (-1)^k\,(k+1)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\frac{1}{(1-x)^2} &= (-1)^0(0+1)(-x)^0 + (-1)^1(1+1)(-x)^1 + \cdots (-1)^k(k+1)(-x)^k + \cdots \\
&= 1 + 2x + 3x^2 + \cdots + (k+1)x^k + \cdots
\end{aligned}
$$

Problem 17 *Find the coefficient of x^k in $\sqrt{1+x} = (1+x)^{\frac{1}{2}}$. Obtain the power series expansion.*

The coefficient of $x^k, k \geq 0$, where k is an integer, in the power series expansion of $(1+x)^n$ is

$$\frac{1}{k!}n(n-1)\cdots(n-k+1)$$

Here $n = \frac{1}{2}$. Therefore, the coefficient of x^k is obtained as follows. In the middle of the computation, we multiply both numerator and denominator by $(k-1)!$. We also multiply and divide by 2^{k-1}. This is followed by multiplication of each individual element of $(k-1)!$ by a 2. Following this, we rearrange the numbers being multiplied to obtain $(2k-2)!$.

$$
\begin{aligned}
& \frac{1}{k!}\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)\cdots\left(\frac{1}{2}-k+1\right) \\
= & \frac{1}{k!}\frac{1}{2}\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)\cdots\left(-\frac{2k-3}{2}\right) \\
= & \frac{(-1)^{k-1}}{k!}\frac{1}{2}\frac{1}{2}\frac{3}{2}\frac{5}{2}\cdots\frac{2k-3}{2} \\
= & \frac{(-1)^{k-1}}{2^k\ k!}1\times 3\times 5\cdots\times(2k-3) \\
= & \frac{(-1)^{k-1}}{2^k\ k!\ (k-1)!}1\times 3\times 5\cdots\times(2k-3)\times(k-1)! \\
= & \frac{(-1)^{k-1}}{2^{k-1}\ 2^k\ k!\ (k-1)!}1\times 3\times 5\cdots\times(2k-3)\times(k-1)!\times 2^{k-1} \\
= & \frac{(-1)^{k-1}}{2^{2k-1}\ k!\ (k-1)!}1\times 3\times 5\cdots(2k-3)\times 2\times 4\times 6\cdots(2k-2) \\
= & \frac{(-1)^{k-1}}{2^{2k-1}\ k!\ (k-1)!}1\times 2\times 3\times 4\cdots\times(2k-2) \\
= & \frac{(-1)^{k-1}}{2^{2k-1}\ k!\ (k-1)!}(2k-2)! \\
= & \frac{2(-1)^{k-1}}{4^k\ k}\frac{(2k-2)!}{(k-1)!\ (k-1)!} \\
= & \frac{2(-1)^{k-1}}{4^k\ k}\binom{2k-2}{k-1}
\end{aligned}
$$

This gives a compact expression for the coefficient of x^k. Since k occurs in the denominator, $k \geq 1$. When $k = 0$, the value of the coefficient x^0 is 1.

5.2 Solving Recurrence Relations Using Generating Functions

In using generating functions, we solve the problem in a roundabout way. A generating function is an infinite series. An *ordinary* generating function that can be used to solve many recurrence relations is the following.

$$
\begin{aligned}
g(x) &= \sum_{n=0}^{\infty} T(n)\ x^n \\
&= T(0)\ x^0 + T(1)\ x^1 + \cdots + T(n)\ x^n + \cdots
\end{aligned} \tag{5.15}
$$

It is a power series whose coefficients are successive terms of the recurrence relation we are trying to solve. The following steps are routinely followed in solving a problem using the generating function method.

1. We write the expression for $g(x)$ as given and manipulate the right hand side.
2. We take one or more terms from the beginning of the series on the right hand side and write them out separately. The number of terms taken out depends on the recurrence relation we need to solve. The rest of the terms are still inside the summation.
3. Next, we use the recurrence relation to substitute the value of $T(n)$ inside the summation.
4. We perform algebraic manipulations to obtain the value of the generating function $g(x)$ as an expression in x. The expression is written as the sum of one or more power series expansions in x.
5. The value of the coefficients of x^n are compared on the left and right hand sides to obtain the nth term in the recurrence relation. This gives us the closed form formula for $T(n)$.

We solve several recurrence relations in the rest of the Chapter.

5.3 Solving the Recurrence $T(n) = T(n-1) + c$

The first recurrence we solve is given below.

$$\begin{aligned} T(n) &= T(n-1) + c & n \geq 1 \\ &= d & n = 0 \end{aligned} \tag{5.16}$$

Here, c and d are small integer constants that are positive. We follow the general steps outlined earlier to solve the recurrence relation.
We write the ordinary generating function for the recurrence relation.

$$\begin{aligned} g(x) &= \sum_{n=0}^{\infty} T(n)x^n \\ &= T(0)x^0 + T(1)x^1 + T(2)x^2 + \cdots \end{aligned}$$

This is a power series whose coefficients are successive terms of the recurrence relation we are trying to solve. The recurrence relation contains one $T()$ term on the right hand side. The coefficient of $T()$ on the right hand side is one less than the coefficient of $T()$ on the left hand side. Because of this, we take the first term off of the infinite series, on the right hand side.

$$= \sum_{n=0}^{\infty} T(n)x^n$$

$$
\begin{aligned}
&= T(0)x^0 + \sum_{n=1}^{\infty} T(n)x^n \\
&= d + \sum_{n=1}^{\infty} T(n)x^n
\end{aligned}
$$

At this point, we use the recurrence relation to expand $T(n)$ inside the summation. The recurrence says $T(n) = T(n-1) + c$. We also separate out the two summations.

$$
\begin{aligned}
&= d + \sum_{n=1}^{\infty} \left[T(n-1) + c\right] x^n \\
&= d + \sum_{n=1}^{\infty} T(n-1)x^n + c\sum_{n=1}^{\infty} x^n
\end{aligned}
$$

There are two summations on the right hand side. In the first summation, the coefficient of $T(n-1)$ is x^n. That is, the argument of $T()$ is one less than the power to which x is raised. To make both the same, we can take an x outside the summation. Since x does not depend on n, the index of the summation, this does not create any problems.

$$
= d + x\sum_{n=1}^{\infty} T(n-1)x^{n-1} + c\sum_{n=1}^{\infty} x^n
$$

Continuing again with the first summation, we see that the index of summation starts at $n = 1$ and goes up to ∞. The argument of $T()$ and the corresponding power to which x is raised are both $n-1$. We can simply change the index to start from $n = 0$, and write the argument of $T()$ and the power to which x is raised as n. If there is any difficulty in understanding, one can write out individual terms of the summation to verify the manipulation.

$$
= d + x\sum_{n=0}^{\infty} T(n)x^n + c\sum_{n=1}^{\infty} x^n
$$

The first sum now is simply the generating function $g(x)$.

$$
= d + x\ g(x) + c\sum_{n=1}^{\infty} x^n
$$

The summation that is left sums x^n from $n = 1$ to ∞. We can write $\sum_{n=1}^{\infty} x^n = \sum_{n=0}^{\infty} x^n - x^0 = \sum_{n=0}^{\infty} x^n - 1$. This is because, we know from our discussions earlier that $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$. We next solve for $g(x)$.

$$
\begin{aligned}
g(x) &= d + x\ g(x) + c\left[\sum_{n=0}^{\infty} x^n - 1\right] \\
&= d + x\ g(x) + c\left[\frac{1}{1-x} - 1\right]
\end{aligned}
$$

$$\begin{aligned}(1-x)\,g(x) &= d + \frac{c}{1-x} - c \\ g(x) &= \frac{d-c}{1-x} + \frac{c}{(1-x)^2}\end{aligned}$$

From the definition of the generating function $g(x)$, we know that the coefficient of x^n in its power series expansion is $T(n)$. We know that the coefficient of x^n in the power series expansion of $\frac{1}{1-x}$ is 1. Therefore, the coefficient of x^n in the power series expansion of $\frac{d}{1-x}$ is d. We also know that the coefficient of x^n in the power series expansion of $\frac{1}{(1-x)^2}$ is $n+1$. Therefore, the coefficient of x^n in the power series expansion of $\frac{c}{(1-x)^2}$ is $c(n+1)$. Now, comparing the coefficients of x^n on both sides of the equation, we get the following.

$$\begin{aligned}T(n) &= c(n+1) + d - c \\ &= cn + c + d - c \\ &= cn + d \end{aligned} \tag{5.17}$$

Thus, the initial assumption relating the coefficient of a power of x, say, $x^n, n \geq 0$ in the generating function to the value of the function $T()$ for argument n, i.e., the value of $T(n)$, helps us in obtaining the general expression for $T(n)$ for the recurrence relation. In other words, the relationship between the generating function and the function $T()$ represented by the recurrence relation was set up expressly for this desired outcome. Thus, the solution to the recurrence relation comes in a slightly circuitous manner, through the medium of the generating function. We note that the power series expansion for fractions of the form $\frac{1}{(1-ax)^i}, i \geq 0, 1-ax \neq 0$, plays a crucial role in solving recurrence relations using generating functions.

The overall approach starts out with the generating function in which the coefficient of $x^n, n \geq 0$ is $T(n)$. Of course, $T(n)$ is unknown at this time and our goal is to find it. By using the recurrence relation and some algebraic manipulations, we express $g(x)$ in such a fashion that the coefficient of x^n becomes known. This is the solution to our recurrence relation.

5.4 Solving the Recurrence: $T(n) = T(n-1) + cn$

The second recurrence relation we solve is given below.

$$\begin{aligned}T(n) &= T(n-1) + cn & n \geq 1 \\ &= d & n = 0\end{aligned} \tag{5.18}$$

Here, c and d are small integer constants that are positive. Once again, we follow the general steps given earlier to solve the recurrence relation.

First, we write the ordinary generating function for the recurrence relation.

$$g(x) = \sum_{n=0}^{\infty} T(n)\, x^n \tag{5.19}$$

This is a power series where the coefficient of x^n is the value of the function $T()$, defined by the recurrence relation, for an argument value of n. The recurrence relation contains $T(n)$ on the left hand side, and one term containing $T()$, namely, $T(n-1)$, on the right hand side. The argument of $T()$ on the right hand side is one less that the argument to $T()$ on the left hand side. As a result, we take the first term off the sum and write it separately. Next, we use the recurrence relation inside the summation, and write the individual sums separately.

$$\begin{aligned}
&= T(0)x^0 + \sum_{n=1}^{\infty} T(n)\, x^n \\
&= d + \sum_{n=1}^{\infty} T(n)\, x^n \\
&= d + \sum_{n=1}^{\infty} [T(n-1) + cn]\, x^n \\
&= d + \sum_{n=1}^{\infty} T(n-1)x^n + c \sum_{n=1}^{\infty} n\, x^n
\end{aligned}$$

We repeat the steps of the previous problem: factor one x out from the first summation, change the lower limit of the summation, change the argument to $T()$ and the power of x. Then, we observe that the first sum actually is the generating function $g(x)$ we started out with.

$$\begin{aligned}
&= d + x \sum_{n=1}^{\infty} T(n-1)\, x^{n-1} + c \sum_{n=1}^{\infty} n\, x^n \\
&= d + x \sum_{n=0}^{\infty} T(n)\, x^n + c \sum_{n=1}^{\infty} n\, x^n \\
&= d + x\, g(x) + c \sum_{n=1}^{\infty} n\, x^n.
\end{aligned}$$

Now, we need to find a closed form expression for $\sum_{n=1}^{\infty} nx^n$. This closed form is going to be in the form of a function of x. We examine the fractions we discussed in the beginning of the Chapter and see that the coefficient of x^{n-1} in the power series expansion for $\frac{1}{(1-x)^2}$ is nx^{n-1}. If we multiply both sides by x, we see that the coefficient of x^n in the power series expansion for $\frac{x}{(1-x)^2}$ is nx^n. Thus, $\sum_{n=0}^{\infty} nx^n = \frac{x}{(1-x)^2}$. Now, we are ready to simplify the summation.

$$\begin{aligned}
&= d + x\, g(x) + c \left[\sum_{n=0}^{\infty} n\, x^n - 0 \right] \\
&= d + x\, g(x) + c \sum_{n=0}^{\infty} n\, x^n
\end{aligned}$$

$$\begin{aligned} &= d + x\, g(x) + \frac{cx}{(1-x)^2} \\ (1-x)\, g(x) &= d + \frac{cx}{(1-x)^2} \\ g(x) &= \frac{d}{1-x} + \frac{cx}{(1-x)^3} \end{aligned}$$

At this point, we have a new expression for $g(x)$. It is not the power series we started with, but a sum of two fractions. We solve the recurrence relation by comparing the coefficients of x^n on both sides. The coefficient of x^n on the right hand side, i.e., in the generating function, is $T(n)$, the value of the function $T()$ defined by the recurrence relation for an argument value of n. The coefficient of x^n in the power series expansion of $\frac{d}{1-x}$ is d. The coefficient of x^{n-1} in the power series for $\frac{1}{(1-x)^3}$ is $\frac{1}{2}n(n+1)$. Therefore, the coefficient of x^n in $\frac{cx}{(1-x)^3}$ is $\frac{1}{2}cn(n+1)$. As a result, we can write the following.

$$\begin{aligned} T(n) &= d + \frac{1}{2}cn(n+1) \\ &= \frac{1}{2}cn^2 + \frac{1}{2}cn + d \end{aligned} \tag{5.20}$$

5.5 Solving the Recurrence $T(n) = 2T\left(\frac{n}{2}\right) + cn$

The recurrence we want to solve next is given as follows.

$$\begin{aligned} T(n) &= 2\,T\left(\tfrac{n}{2}\right) + cn && n \geq 1 \\ &= d && n = 1 \end{aligned} \tag{5.21}$$

A recurrence such as this is not usually solvable for all positive integer values n. To solve the recurrence relation, we need to make an assumption regarding the value of n. The assumption that we make is $n = 2^m$ for some $m \geq 0$. With this assumption, we can rewrite the recurrence as follows.

$$\begin{aligned} T(2^m) &= 2\,T(2^{m-1}) + c\,2^m && m \geq 1 \\ T(2^0) &= d && m = 0 \end{aligned} \tag{5.22}$$

We now have changed the recurrence relation from one in terms of n to one in terms of m. This is variable substitution. The original termination condition is given $n = 1$. In terms of m, the termination condition becomes $m = 0$. In addition, we also perform a function substitution. Let us define a new function ϕ such that

$$\phi(m) = T(2^m) \tag{5.23}$$

With this assumption, we can rewrite the recurrence relation as the following.

$$\begin{aligned} \phi(m) &= 2\,\phi(m-1) + c\,2^m && m \geq 1 \\ \phi(0) &= d && m = 0 \end{aligned} \tag{5.24}$$

We solve the last recurrence relation using a generating function. Once we have solved it, we can perform function and variable substitutions in reverse to get the solution to the original recurrence relation. Let

$$g(x) = \sum_{k=0}^{\infty} \phi(k)\ x^k \tag{5.25}$$

be the generating function for the last recurrence relation. Here, the coefficient for x^k is $\phi(k)$, the value of the function $\phi()$ in the recurrence relation for the argument value k. We perform the steps that we carried out in the previous two problems. We take one term out of the summation, substitute the recurrence relation for $\phi(k)$, separate out the sums, change the limits in the first sum, and express the second sum in terms of a fraction, $\frac{1}{1-2x}$. The steps are given below.

$$\begin{aligned}
g(x) &= \sum_{k=0}^{\infty} \phi(k)\ x^k \\
&= \phi(0)\ x^0 + \sum_{k=1}^{\infty} \phi(k)\ x^k \\
&= d + \sum_{k=1}^{\infty} \left[2\ \phi(k-1) + c\ 2^k\right] x^k \\
&= d + 2\sum_{k=1}^{\infty} \phi(k-1)\ x^k + c\sum_{k=1}^{\infty} (2x)^k \\
&= d + 2x\sum_{k=1}^{\infty} \phi(k-1)\ x^{k-1} + c\left[\sum_{k=0}^{\infty} (2x)^k - 1\right] \\
&= d + 2x\sum_{k=0}^{\infty} \phi(k)\ x^k + c\sum_{k=0}^{\infty} (2x)^k - c \\
&= d + 2x\ g(x) + c\sum_{k=0}^{\infty} (2x)^k - c
\end{aligned}$$

We need to find a closed form fraction for $\sum_{k=0}^{\infty}(2x)^k$. If we look at the fractions we discussed in the beginning of the Chapter, we see that $\frac{1}{1-ax} = \sum_{k=0}^{\infty}(ax)^k, 1 - ax \neq 0$. If we take a's value to be 2, we see that $\frac{1}{1-2x} = \sum_{k=0}^{\infty}(2x)^k$. Continuing with our solution, we get the following.

$$\begin{aligned}
g(x) &= d + 2x\ g(x) + \frac{c}{1-2x} - c \\
(1-2x)\ g(x) &= (d-c) + \frac{c}{1-2x} \\
g(x) &= \frac{d-c}{1-2x} + \frac{c}{(1-2x)^2}
\end{aligned}$$

Now, we need to compare the coefficients of $x^m, m \geq 0$ on both sides. The coefficient of x^m in the infinite series for $\frac{1}{1-2x}$ is 2^m. The coefficient of x^m in the infinite series for $\frac{1}{(1-2x)^2}$ is $(m+1)2^m$. Therefore, we get the following.

$$\begin{aligned} \phi(m) &= (d-c)2^m + c(m+1)2^m \\ &= (d-c+cm+c)2^m \\ &= (cm+d)2^m \end{aligned}$$

We now remember that we had performed a function substitution by assuming $\phi(m) = T(2^m)$. We perform the substitution in the reverse to obtain the following.

$$T(2^m) = (cm+d)2^m$$

Now, we had performed a variable substitution earlier: $n = 2^m, m \geq 0$. This also tells us that $m = log_2\, n$. We perform the substitution in reverse to get the following.

$$\begin{aligned} T(n) &= (c\, log_2\, n + d)n \\ &= cn\, log_2\, n + dn \end{aligned} \quad (5.26)$$

5.6 Solving the Recurrence $T(n) = T(n-1) + T(n-2)$

The recurrence we solve next is given below.

$$\begin{aligned} T(n) &= T(n-1) + T(n-2) & n \geq 2 \\ T(0) &= 1 & n = 0 \\ T(1) &= 1 & n = 1 \end{aligned} \quad (5.27)$$

This recurrence relation defines the *Fibonnaci numbers*. Fibonnaci numbers occur quite frequently in science and engineering, including computer science. A Fibonnaci number is obtained by adding the previous two Fibonacci numbers. The first two Fibonacci numbers in the sequence are given as $T(0) = T(1) = 1$.

The recurrence relation given for Fibonacci numbers is linear, but it is more complex than the ones we have solved so far. The relation is linear because we have two linear terms: $T(n-1)$ and $T(n-2)$ on the right hand side, and no higher degree terms. It is more complex than the previous recurrences because we have two $T()$ terms on the right hand side. Each recurrence we solved earlier has only one $T()$ term on the right.

However, the problem is not much more difficult to solve than the recurrences we have solved earlier. Successive Fibonacci numbers are coefficients of increasing powers of x in the generating function. Since there are two $T()$ terms on the right hand side, we write the first two terms of the summation separately. In the summation that is left, we substitute $T(n)$ by the recurrence $T(n-1) + T(n-2)$. We perform algebraic manipulation to obtain a new expression in x, for the generating function. The algebraic manipulation involves changing the lower limit of the two summations.

$$\begin{aligned}
g(x) &= \sum_{n=0}^{\infty} T(n)\, x^n \\
&= T(0)\, x^0 + T(1)\, x^1 + \sum_{n=2}^{\infty} T(n)\, x^n \\
&= 1 + x + \sum_{n=2}^{\infty} T(n)\, x^n \\
&= 1 + x + \sum_{n=2}^{\infty} T(n-1)\, x^n + \sum_{n=2}^{\infty} T(n-2)\, x^n \\
&= 1 + x + x\sum_{n=2}^{\infty} T(n-1)\, x^{n-1} + \sum_{n=2}^{\infty} T(n-2)\, x^{n-2} \\
&= 1 + x + x\sum_{n=1}^{\infty} T(n)\, x^n + x^2\sum_{n=0}^{\infty} T(n)\, x^n \\
&= 1 + x + x\left[\sum_{n=0}^{\infty} T(n)\, x^n - T(0)\, x^0\right] + x^2\, g(x) \\
&= 1 + x + x\,[g(x) - 1] + x^2\, g(x) \\
&= 1 + x + x\, g(x) - x + x^2\, g(x) \\
(-x^2 - x + 1)\, g(x) &= 1 \\
g(x) &= \frac{1}{-x^2 - x + 1}
\end{aligned}$$

We halt our progress to a solution here for the time being to obtain partial fractions for the right hand side. At this point, we have a fractional expression for $g(x)$. The denominator of the fraction is quadratic. First, we check if the quadratic equation has non-complex roots. For this, we compute $b^2 - 4ac$ where a, b and c are the coefficients of x^2, x^1 and x^0, respectively, in the quadratic equation.

$$b^2 - 4ac = (-1)^2 - 4(-1)1 = 1 + 4 = 5$$

5 is positive, and hence, the quadratic equation has two rational roots. As a result, we can obtain two linear factors. Let us find the two roots of $-x^2 - x + 1$.

$$\begin{aligned}
The\ two\ roots &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\
&= \frac{-(-1) \pm \sqrt{5}}{2 \times (-1)} \\
&= \frac{1 \pm \sqrt{5}}{-2}
\end{aligned}$$

Now, let us define two constants α_1 and α_2.

$$\begin{aligned}\alpha_1 &= \frac{1+\sqrt{5}}{2} = 1.618034 \\ \alpha_2 &= \frac{1-\sqrt{5}}{2} = -0.618304\end{aligned}$$

We need to obtain linear factors for $-x^2 - x + 1$. For this, we observe the following.

$$\alpha_1\ \alpha_2 = \frac{1+\sqrt{5}}{2} \times \frac{1-\sqrt{5}}{2} = \frac{1-5}{4} = \frac{-4}{4} = -1$$

and

$$\alpha_1 + \alpha_2 = \frac{1+\sqrt{5}}{2} + \frac{1-\sqrt{5}}{2} = \frac{2}{2} = 1$$

Therefore, we can factorize as follows.

$$\begin{aligned}-x^2 - x + 1 &= \alpha_1\ \alpha_2\ x^2 - (\alpha_1 + \alpha_2)x + 1 \\ &= \alpha_1\ \alpha_2\ x^2 - \alpha_1\ x - \alpha_2\ x + 1 \\ &= \alpha_1\ x\ (\alpha_2\ x - 1) - 1(\alpha_2\ x - 1) \\ &= \alpha_1\ x(\alpha_2\ x - 1) - 1(\alpha_2\ x - 1) \\ &= (\alpha_1\ x - 1)(\alpha_2\ x - 1) \\ &= (1 - \alpha_1\ x)(1 - \alpha_2\ x)\end{aligned}$$

Now, we need to obtain partial fractions for $\frac{1}{-x^2-x+1}$ with linear polynomials in the denominator. Let the fractions be $\frac{A}{1-\alpha_1\ x}$ and $\frac{B}{1-\alpha_2\ x}$ where A and B are constants to be determined. Let

$$\frac{1}{-x^2 - x + 1} = \frac{1}{(1-\alpha_1\ x)(1-\alpha_2\ x)} = \frac{A}{1-\alpha_1\ x} + \frac{B}{1-\alpha_2\ x}$$

Multiplying both sides by $(1 - \alpha_1\ x)(1 - \alpha_2\ x)$, we get

$$1 = A(1 - \alpha_2\ x) + B(1 - \alpha_1 x)$$

If we let $x = \frac{1}{\alpha_2}$, we get the following.

$$\begin{aligned}1 &= A\left(1 - \alpha_2\frac{1}{\alpha_2}\right) + B\left(1 - \alpha_1\frac{1}{\alpha_2}\right) \\ &= B\left(1 - \frac{\alpha_1}{\alpha_2}\right) \\ &= B\frac{\alpha_2 - \alpha_1}{\alpha_2} \\ &= B\frac{\frac{1-\sqrt{5}}{2} - \frac{1+\sqrt{5}}{2}}{\alpha_2}\end{aligned}$$

$$
\begin{aligned}
&= \frac{B}{2\,\alpha_2}(1-\sqrt{5}-1-\sqrt{5}) \\
&= -\frac{B\sqrt{5}}{\alpha_2} \\
B &= -\frac{\alpha_2}{\sqrt{5}}
\end{aligned}
$$

If we let $x = \frac{1}{\alpha_1}$, we get the following.

$$
\begin{aligned}
1 &= A\left(1-\frac{\alpha_2}{\alpha_1}\right)+0 \\
&= A\frac{\alpha_1-\alpha_2}{\alpha_1} \\
&= \frac{A}{\alpha_1}\left(\frac{1+\sqrt{5}}{2}-\frac{1-\sqrt{5}}{2}\right) \\
&= \frac{A\sqrt{5}}{\alpha_1} \\
A &= \frac{\alpha_1}{\sqrt{5}}
\end{aligned}
$$

Now, we resume solving the recurrence.

$$
\begin{aligned}
g(x) &= \frac{1}{-x^2+x+1} \\
&= \frac{\alpha_1}{\sqrt{5}(1-\alpha_1\,x)} - \frac{\alpha_2}{\sqrt{5}(1-\alpha_2\,x)}
\end{aligned}
$$

We compare the coefficients of x^n on the left hand side and the right hand side. The coefficient of x^n in the power series expansion of $g(x)$ is $T(n)$. The coefficient of x^n in the power series expansions for $\frac{1}{1-\alpha_1\,x}$ and $\frac{1}{1-\alpha_2\,x}$ are α_1^n and α_2^n, respectively. Therefore, we can write the expression for $T(n)$ as follows.

$$
\begin{aligned}
T(n) &= \frac{\alpha_1}{\sqrt{5}}\alpha_1^n - \frac{\alpha_2}{\sqrt{5}}\alpha_2^n \\
&= \frac{1}{\sqrt{5}}\left[\alpha_1^{n+1}-\alpha_2^{n+1}\right] \\
&= \frac{1}{\sqrt{5}}\left[1.618034^{n+1}-(-0.618034)^{n+1}\right] \qquad (5.28)
\end{aligned}
$$

If n is large, we can write

$$
T(n) \approx \frac{1}{\sqrt{5}}\alpha_1^{n+1} = \frac{1}{\sqrt{5}}1.618034^{n+1} \qquad (5.29)
$$

Even for a relatively small value of n such as 10, the approximation works pretty well. The following table shows values of Fibonacci numbers obtained for a few values of n using the exact and the approximate formulas.

$T(n)$	$\frac{1}{\sqrt{5}}\left(\alpha_1^{n+1} - \alpha_2^{n+1}\right)$	$\frac{1}{\sqrt{5}}\alpha_1^{n+1}$
1	1.00000000000000000000	1.17082039324993690891
2	1.99999999999999999997	1.89442719099991587854
3	2.99999999999999999995	3.06524758424985278743
10	88.9999999999999999956	88.9977527522458149541
20	10945.9999999999999989	10945.9999817284852598
50	20365011073.9999999950	20365011073.9999999950

5.7 Solving the Recurrence $a_n = \sum_{i=0}^{n} a_i a_{n-i}$

In this section, we solve a recurrence that arises in several combinatorial situations. One situation where this recurrence arises is when we want to obtain the number of ways, a_n, in which we can place parentheses to multiply n numbers, $k_1, k_2, \cdots, k_n$, when we are constrained to multiply only two numbers at a time, like when we use a calculator.

We start with the simplest case. There is only one way, (k_1k_2), to parenthesize and multiply two numbers. Therefore, $a_2 = 1$.

The number of ways to multiply three numbers, k_1, k_2 and k_3 is two: $(k_1k_2)k_3$ and $k_1(k_2k_3)$. Therefore, we conclude $a_3 = 2$.

We do not know what a_0 and a_1 should be. We simply assume $a_0 = 0$ and $a_1 = 1$ to solve the recurrence.

If we are given n numbers, $k_1, k_2, \cdots, k_n$, we can partition them into two parts: k_1 through $k_i, i \geq 1$, and k_{i+1} through k_n. We obtain the products $k_1k_2\cdots k_i$ and $k_{i+1}k_{i+2}\cdots k_n$, by using any parenthesization we want, and then multiply the two sub-products to obtain the final product. That is, the product is obtained as follows.

$$((k_1k_2\cdots k_i)(k_{i+1}\cdots k_n))$$

The first sub-product multiplies i numbers, and the second sub-product multiplies $n - i$ numbers. So, the first sub-product can be obtained in a_i ways, and the second sub-product can be obtained in a_{n-i} ways. Therefore, the product $((k_1k_2\cdots k_i)(k_{i+1}\cdots k_n))$ can be obtained in $a_n a_{n-i}$ ways. Finally, we note that the final product can be obtained for every possible value of i from 1 to $n-1$. All of the following parenthesizations are possible.

$$\begin{array}{l} ((k_1)(k_2k_3\cdots k_{n-1}k_n)) \\ ((k_1k_2)(k_3\cdots k_{n-1}k_n)) \\ \vdots \\ ((k_1k_2\cdots k_{n-2})(k_{n-1}k_n)) \\ ((k_1k_2k_3\cdots k_{n-1})(k_n)) \end{array}$$

Thus, the total number of ways in which the final product can be obtained is given as:

$$a_n = \sum_{i=1}^{n-1} a_i \; a_{n-i}$$

Since we have assumed $a_0 = 0$ and $a_1 = 1$, we have $a_0 a_n = 0$, and therefore, we can write the recurrence as

$$\begin{aligned} a_n &= \sum_{i=0}^{n} a_i \; a_{n-i} \\ &= a_0 \; a_n + a_1 \; a_{n-1} + \cdots + a_{n-1} \; a_1 + a_n \; a_0 \end{aligned} \tag{5.30}$$

We can solve for a_n using an ordinary generating function. Let $g(x)$ be the generating function. We take two terms out of the summation on the right hand side and then use the recurrence relation to substitute for a_n.

$$\begin{aligned} g(x) &= \sum_{n=0}^{\infty} a_n \; x^n \\ &= a_0 \; x^0 + a_1 \; x^1 + \sum_{n=2}^{\infty} a_n \; x^n \\ &= 0 + x + \sum_{n=2}^{\infty} \left(\sum_{i=0}^{n} a_i \; a_{n-i} \right) x^n \\ &= x + \sum_{n=2}^{\infty} \left(\sum_{i=0}^{n} a_i \; a_{n-i} \right) x^n \end{aligned}$$

At this point, we stop the solution to the recurrence temporarily in order to find an expression in terms of $g(x)$ for the double summation on the right hand side. It looks complex, but some observations will make it easy to obtain.
We start by computing $(g(x))^2$.

$$\begin{aligned} (g(x))^2 &= \left(\sum_{n=0}^{\infty} a_n x^n \right)^2 \\ &= (a_0 \; x^0 + a_1 \; x^1 + \cdots + a_{n-1} \; x^{n-1} + a_n \; x^n + \cdots)^2 \\ &= (a_0 \; x^0 + a_1 \; x^1 + \cdots + a_{n-1} \; x^{n-1} + a_n \; x^n + \cdots)(a_0 \; x^0 + a_1 \; x^1 + \cdots + a_{n-1} \; x^{n-1} + a_n \; x^n + \cdots) \end{aligned}$$

x^n is obtained on the right hand side in n ways. For example, when we take $a_0 x^0$ from the first parenthesized expression and multiply it by $a_n x^n$ from the second parenthesized expression, we get a term containing x^n. We also get a term containing x^n when we multiply $a_1 x^1$ from the first expression by $a_{n-1} x^{n-1}$ from the second expression. Considering all the multiplications that produce x^n, we get the expression for the coefficient of x^n in $(g(x))^2$

as follows.

$$\begin{aligned} & a_0\ a_n + a_1\ a_{n-1} + \cdots + a_{n-1}\ a_1 + a_n\ a_0 \\ = & \sum_{i=0}^{n} a_i\ a_{n-i} \end{aligned}$$

Therefore, we can write $(g(x))^2$ as follows.

$$\begin{aligned} (g(x))^2 &= a_0\ a_0\ x^0 + (a_0\ a_1 + a_1\ a_0)x^1 + \sum_{n=2}^{\infty}\left(\sum_{i=0}^{n} a_i\ a_{n-i}\right) x^n \\ &= 0 + 0 + \sum_{n=2}^{\infty}\left(\sum_{i=0}^{n} a_i\ a_{n-i}\right) x^n \\ &= \sum_{n=2}^{\infty}\left(\sum_{i=0}^{n} a_i\ a_{n-i}\right) x^n \end{aligned}$$

Now, we can get back to the original recurrence relation we are solving. We had stopped mid-way to obtain an expression, in terms of $g(x)$, for $\sum_{n=2}^{\infty}\left(\sum_{i=0}^{n} a_i a_{n-i}\right) x^n$. We have that expression now: $(g(x))^2$. Therefore, we can write

$$\begin{aligned} g(x) &= x + (g(x))^2 \\ ((g(x))^2 - g(x) + x &= 0 \end{aligned}$$

This is a functional equation for $g(x)$. In particular, it is a quadratic equation in $g(x)$, with coefficients $1, -1$ and x, respectively. Therefore, we can use the formula for roots of quadratic equations to write:

$$g(x) = \frac{1 \pm \sqrt{1-4x}}{2}$$

There are two solutions: $\frac{1+\sqrt{1-4x}}{2}$ and $\frac{1-\sqrt{1-4x}}{2}$. We can use either one of these solutions, but the one that is strictly correct for our purpose is the one that makes $g(x)$ assume the correct value at a value of x for which $g(x)$ is known. In particular, at $x = 0$,

$$g(0) = \sum_{n=0}^{\infty} a_n\ 0^n = 0$$

because 0 raised to any power is zero. The solution to the quadratic equation that gives $g(0) = 0$ is $\frac{1-\sqrt{1-4x}}{2}$. Therefore, we can continue with our solution for the recurrence by comparing the coefficients of x^n on both sides of $\frac{1-\sqrt{1-4x}}{2}$. We use the Generalized Binomial Theorem to obtain the coefficients of x^n in $(1-4x)^{\frac{1}{2}}$ that is given directly below.

$$= \frac{1}{n!}\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)\cdots\left(\frac{1}{2}-n+1\right)(-4)^n$$

$$\begin{aligned}
&= \frac{1}{n!}\frac{1}{2}\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)\cdots\left(-\frac{2n-3}{2}\right)(-4)^n \\
&= \frac{1}{n!}\frac{1\times 3\times 5\cdots\times(2n-3)}{2^n}(-1)^{n-1}(-4)^n \\
&= -\frac{1}{n!}1\times 3\times 5\cdots\times(2n-3)\times 2^n \\
&= -\frac{1}{n!}\frac{1\times 3\times 5\cdots\times(2n-3)}{(n-1)!}(n-1)!\times 2^n \\
&= -\frac{1}{n!\,(n-1)!}1\times 3\times 5\cdots\times(2n-3)\times 1\times 2\times 3\cdots(n-1)\times 2^{n-1}\times 2 \\
&= -\frac{1}{n!\,(n-1)!}1\times 3\times 5\cdots\times(2n-3)\times 2\times 4\times 6\cdots(2n-2)\times 2 \\
&= -\frac{1}{n!\,(n-1)!}1\times 2\times 3\times 4\cdots\times(2n-2)\times 2 \\
&= -\frac{2(2n-2)!}{n!\,(n-1)!} \\
&= -\frac{2}{n}\frac{(2n-2)!}{(n-1)!\,(n-1)!} \\
&= -\frac{2}{n}\begin{pmatrix} 2n-2 \\ n-1 \end{pmatrix}
\end{aligned}$$

The last step is obtained because we know

$$\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n!}{(n-k)!\,k!}$$

Thus, the above gives

$$a_n = \frac{1}{n}\begin{pmatrix} 2n-2 \\ n-1 \end{pmatrix} \quad (5.31)$$

as solution to the recurrence.

We can perform some additional computation using what is known as *Stirling's formula* that gives us a way to compute $n!$. Stirling's formula is very well-known and is given as

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right) \quad (5.32)$$

The term $\Theta\left(\frac{1}{n}\right)$ is tightly bound by a multiple of $\frac{1}{n}$ both from above and below. If n is large, we can ignore this term and write:

$$\begin{aligned}
n! &\approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \\
&= \sqrt{2\pi}\, n^{n+\frac{1}{2}}\, e^{-n} \quad (5.33)
\end{aligned}$$

Now, we can simplify the formula for a_n.

$$\begin{aligned}
a_n &= \frac{1}{n}\binom{2n-2}{n-1} \\
&= \frac{1}{n}\frac{(2n-2)!}{(n-1)!\,(n-1)!} \\
&= \frac{(2n-2)!}{n!\,(n-1)!} \\
&= \frac{\sqrt{2\pi}\,(2n-2)^{2n-2+\frac{1}{2}}\,e^{-2n+2}}{\sqrt{2\pi}\,n^{n+\frac{1}{2}}\,e^{-n}\,\sqrt{2\pi}\,(n-1)^{n-1+\frac{1}{2}}\,e^{-n+1}} \\
&\approx \frac{(2n)^{2n-\frac{3}{2}}\,e}{\sqrt{2\pi}\,n^{n+\frac{1}{2}}\,n^{n-\frac{1}{2}}} \\
&= \frac{2^{2n-\frac{3}{2}}\,n^{2n-\frac{3}{2}}\,e}{\sqrt{2\pi}\,n^{2n}} \\
&= \frac{2^{2n}\,n^{-\frac{3}{2}}\,e}{2^{\frac{3}{2}}\,\sqrt{2\pi}} \\
&= \frac{4^n\,e}{4\,\sqrt{\pi}\,n^{\frac{3}{2}}} \\
&= \frac{4^{n-1}\,e}{\sqrt{\pi}\,n^{\frac{3}{2}}} \qquad (5.34)
\end{aligned}$$

This is the final solution to the original recurrence relation.

Bibliography

[BV00] Sara Baase and Allen Van Gelder. *Computer Algorithms: Introduction to Design and Analysis*. Addison-Wesley, Reading, Massachusetts, third edition, 2000.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, second edition, 2001.

[GK90] Daniel Greene and Donald E. Knuth. *Mathematics for the Analysis of Algorithms*. Birkh´auser, third edition, 1990.

[GT02] Michael T. Goodrich and Roberto Tamassia. *Algorithm Design: Foundations, Analysis and Internet Examples*. John Wiley & Sons, Danvers, Massachusetts, 2002.

[Knu98] Donald E. Knuth. *The Art of Computer Programming*. Addison-Wesley, Boston, Massachusetts, second edition, 1998.

[Lev03] Anany Levitin. *Introduction to the Design and Analysis of Algorithms*. Addison-Wesley, Boston, Massachusetts, 2003.

[LMO85] J.C. Lagarias, V.S. Miller, and A.M. Odlyzko. Computing $\pi(x)$: The missel-lehmer method. *Mathematical Computation*, 44:537–560, 1985.

[McC01] Jeffrey J. McConnell. *Analysis of Algorithms, An Active Learning Approach*. Jones and Bartlett Computer Science, Sudbury, Massachusetts, 2001.

[MvV97] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Louisiana, 1997.

[Sed98] Robert Sedgewick. *Algorithm in C++, Parts 1-4, Fundamentals, Data Structures, Sorting, Searching*. Addison-Wesley, Boston, Massachusetts, third edition, 1998.