Chapter 2

Divide and Conquer

The divide-and-conquer paradigm is based on the following idea: Divide problem into smaller parts of the same type. Find solutions for parts. Combine solutions to give solution to original problem.
Examples: Binary search and Quicksort
Implementations of divide and conquer algorithms are often recursive. The running time functions of divide and conquer algorithms can be written in the following way:

$$T(n) = aT(\frac{n}{b}) + d(n)$$

$$T(1) = 1 \text{ , where}$$

- $T(n)$: amount of work to solve original problem
- a: the number of subproblems we have to solve
- $\frac{n}{b}$: number of parts the problem is divided into
- $d(n)$ is the amount of work required to combine a subproblems of size $\frac{n}{b}$

2.1 Quicksort (partition exchange sort)

Quicksort was invented by *C.A.R. Hoare*. Previous fast sorts had used extra storage.

- **Problem:**
  Given an unsorted array L of size n, we want to sort L in ascending order.
- **Idea:**
  It would be nice, if we could find the median of an array and split the array in two equal parts.
- **Technique:**
  At each step an element of L is placed in its final position in the array. This means that all entries before are smaller, and all entries after it larger. The partition element can be chosen:

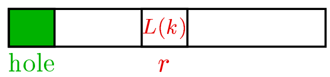
  i) first element in L
  ii) any random element in L
  iii) median of any three elements in L
  iv) middle element of L

In what follows, we choose the partition element according to *iv*). We make use of SPLIT, an algorithm that chooses an element X from the subarray that is currently being sorted and rearranges the entries, finding an integer i such that $L(r) \leq X$ for $k \leq r < i$, $L(i) = X$, and $L(r) \geq X$ for $i < r \leq m$. Thus, X (which is in the i^{th} position) is now in its final correct position and can be ignored.

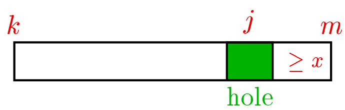
Remark: Holes are really copying operations. Just assume that when we copy we leave a hole. Quicksort is a recursive procedure.

```
QUICKSORT(L, k, m)                    Sort L(k)...L(m)
    begin
        if k < m then do
        begin
            SPLIT(L, k, m, i);        Divide
            QUICKSORT(L, k, i-1);
                                      Conquer
            QUICKSORT(L, i+1, m)
        end
    end
```

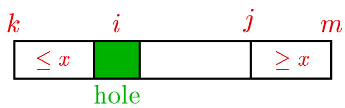
```
SPLIT(L, k, m, i)
    begin
        r:=⌊(k+m)/2⌋;
        X =L(r);
        L(r)=L(k);                          {start with hole at k}
        i:=k;
        j:=m;
        repeat
        begin
```




```
            while (i<j) and (L(j)≥ X) do
                j:=j-1
            if i=j then exit                {from repeat}
            else
                begin
                L(i):=L(j);                 {hole at j}
                i:=i+1
                end
```

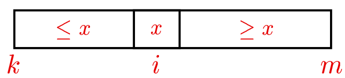



```
            while (i<j) and (L(i)≤ X) do
                i:=i+1
            if i=j then exit                {from repeat}
            else
                begin
                L(j):=L(i);                 {hole at i}
                j:=j-1
                end
```




```
        end                                 {of repeat}
        L(i):=X
    end
```

Array after the execution of SPLIT:



Remark: SPLIT can be written without the 2 exits. But SPLIT as written here is clearer for explaining the steps and analysis.

2.1.1 Worst-case analysis

Occurs when the splitting element X is the largest (or smallest) element in the current subfile $L(k)$, $L(k+1)$, ..., $L(m-1)$, $L(m)$. Thus, SPLIT produces an empty right (or left) subfile. This means that only one **while** loop will be executed each time through the repeat loop.
Passes in repeat loop:

1^{st} pass	n elements	$n-1$ comparisons
2^{st} pass	$n-1$ elements	$n-2$ comparisons
$\vdots$	$\vdots$	$\vdots$
$n-1$ pass	2 elements	1 comparison

$\Rightarrow\ W(n) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$

$\Rightarrow$ Quicksort is $O(n^2)$ in the worst case.

2.1.2 Average case analysis

It can be shown that $AC(n)$ is $O(n \log_2 n)$, where $AC(n)$ denotes the average number of comparisons required for a file of size n. We assume that the element chosen to split is equally likely to end up in each of $L(k)$, $L(k+1)$, ..., $L(m)$.

$$AC(n) = n + 1 + \frac{1}{n}\sum_{i=1}^{n}[AC(i-1) + AC(n-i)]$$

where

$n+1$	1^{st} pass requires at most $n+1$ comparisons (even if implemented in a different way) regardless of where the partition element is
$\frac{1}{n}$	i has equal probability of being $1, 2, \ldots, n$.
$AC(i-1)$	left subarray
$AC(n-i)$	right subarray
$AC(1) = 0$	
$AC(0) = 0$	

$$\begin{aligned}\Rightarrow nAC(n) &= (n+1)n + \tfrac{n}{n}\left[\sum_{i=1}^{n} AC(i-1) + \sum_{i=1}^{n} AC(n-i)\right] \\ &= (n+1)n + \left[\sum_{i=1}^{n} AC(i-1) + \sum_{i=1}^{n} AC(i-1)\right]\end{aligned}$$

$$\Rightarrow [AC(n)]n = (n+1)n + 2\sum_{i=1}^{n} AC(i-1)$$

$$\textbf{Note:}\ \sum_{i=1}^{n} AC(n-i) = AC(n-1) + AC(n-2) + \ldots + AC(0)$$

$$nAC(n) = n(n+1) + 2[AC(0) + AC(1) + \ldots + AC(n-1)] \qquad (1)$$

and

$$(n-1)AC(n-1) = (n-1)n + 2[AC(0) + AC(1) + \ldots + AC(n-2)] \qquad (2)$$

Subtract to obtain (perform (1) − (2)):

$$nAC(n) - (n-1)AC(n-1) = n(n+1) - n(n-1) + 2[AC(n-1)]$$

$$\Rightarrow nAC(n) - (n-1)AC(n-1) = n[n+1-n+1] + 2AC(n-1)$$

$$\Rightarrow nAC(n) = (n-1+2)AC(n-1) + 2n$$

$$nAC(n) = (n+1)AC(n-1) + 2n$$

Multiply both sides by $\frac{1}{n(n+1)}$.

$$\Rightarrow \frac{AC(n)}{n+1} = \frac{AC(n-1)}{n} + \frac{2}{n+1}$$

Thus

$$\begin{array}{ccccc} \frac{AC(n-1)}{n} & = & \frac{AC(n-2)}{n-1} & + & \frac{2}{n} \\ \frac{AC(n-2)}{n-1} & = & \frac{AC(n-3)}{n-2} & + & \frac{2}{n-1} \\ \vdots & \vdots & \vdots & & \\ \frac{AC(2)}{3} & = & \frac{AC(1)}{2} & + & \frac{2}{3} \end{array}$$

Add to get

$$\begin{aligned} \frac{AC(n)}{n+1} &= \frac{AC(1)}{2} + 2\left[\frac{1}{3} + \frac{1}{4} + \ldots + \frac{1}{n+1}\right] \\ &= 2\sum_{i=3}^{n+1} \frac{1}{i} \\ &= 2\left[\sum_{i=1}^{n+1} \frac{1}{i} - \left(1 + \frac{1}{2}\right)\right] \end{aligned}$$

$$\Rightarrow AC(n) = 2(n+1)\left(\sum_{i=1}^{n+1} \frac{1}{i} - \frac{3}{2}\right)$$

$$\Rightarrow AC(n) \text{ is } \Theta(n \log_2 n), \text{ since}$$

$$\sum_{i=1}^{n} \frac{1}{i} \text{ is } \Theta(\log_2 n) \text{ and } \Theta(n) < \Theta(n \log_2 n)$$

□

2.2 General divide&conquer

The general Divide&Conquer procedure, DIV&CONQ makes use of OPERATE, which is a procedure that runs efficiently on small problem instances.

```
DIV&CONQ(X)
    if X is sufficiently small or simple then return OPERATE(X)
    else
        decompose X in smaller instances X_1, X_2, ..., X_2
        for i ← 1 to n
            Y_i ← DIV&CONQ(X_i)
        recombine the Y_i's to obtain a solution Y for X
        return Y
```

2.2.1 Other examples

- **Binary Search**:
  Search a sorted array for a particular "key".
  Idea: Check the middle element of the array; if it equals "key" then we are done. If it is less than "key" then perform a binary search of upper half, otherwise perform a binary search of lower half. **Note**: Only one of the two subproblems is further considered.

$$\text{In general: } T(n) = T\left(\frac{n}{2}\right) + c \quad \text{(n is a power of 2)}$$

  or more precisely:

$$T(n) = \underbrace{T(\left\lfloor \frac{n}{2} \right\rfloor)}_{\substack{\text{Time for}\\ \text{solving}\\ \text{subproblem}\\ \text{of size } \frac{n}{2}}} + \underbrace{1}_{\substack{\text{Amount of}\\ \text{work needed}\\ \text{to split problem}}}$$

- Max-Min **Problem**
  Find the maximum and the minimum values in an unsorted array.
  Idea: Break the array in half. Find the maximum and minimum of each half separately (**Note:** both subproblems are solved here). Then, the minimum is the smaller of the two minima and the maximum is the larger of the two maxima.

$$\text{Here: } T(n) = 2T(\frac{n}{2}) \ + \ c$$

  when n is a power of 2.

- Mergesort **Problem**
  Sort an array in ascending/descending order.
  Idea: Split the array in half and recursively sort each half (using Mergesort). Then merge the two halves with one pass through each half.

$$\text{Here : } T(n) = 2T(\frac{n}{2}) \ + \ cn \text{ when } n = 2^j$$

In the next section, we study a general recurrence relation encountered in the analysis of many algorithms.

2.2.2 The general recurrence relation

Consider:

$$\begin{array}{ll} T(n) = aT(\frac{n}{b}) \ + \ d(n) & n > 1 \\ T(n) = 1 & n = 1 \end{array}$$

We hope that $d(n)$ is a nice function, otherwise we generally approximate it.

Particular Case:
Suppose that $n = b^j$ and $d(n) = cn^i = \Theta(n^i)$. We show that the closed form of

$$\begin{array}{ll} T(n) = aT(\frac{n}{b}) \ + \ d(n) & n > 1 \\ T(n) = 1 & n = 1 \end{array}$$

$$\text{is : } T(b^j) = a^j \ + \ cb^{ji} \sum_{k=0}^{j-1} \left(\frac{a}{b^i}\right)^k \qquad (*)$$

Proof:
Use induction on j
Basis: $j = 0$

$T(b^0) = a^0 + cb^0(0) = a^0 = 1$
which is correct, since we
are given T(1)=1

Induction hypothesis:
Suppose that $(*)$ is true for $j = m - 1$

$$\text{i.e. } T(b^{m-1}) = a^{m-1} + cb^{(m-1)i} \sum_{k=0}^{m-2} \left(\frac{a}{b^i}\right)^k$$

Show that $(*)$ is true for $j = m$.
We know that:

$$T(b^m) = aT(\frac{b^m}{m}) + c(b^m)^i \text{ from the recurrence relation}$$

$$\begin{aligned}
\Rightarrow T(b^m) &= aT(b^{m-1}) \ + \ cb^{mi} \\
&= a\left[a^{m-1} + cb^{(m-1)i} \sum_{k=0}^{m-2} \left(\tfrac{a}{b^i}\right)^k\right] \ + \ cb^{mi} \quad \text{(by induction hypothesis)} \\
&= a^m + acb^{(m-1)i} \sum_{k=0}^{m-2} \left(\tfrac{a}{b^i}\right)^k + cb^{mi} \\
&= a^m + \tfrac{acb^{mi}}{b^i} \sum_{k=0}^{m-2} \left(\tfrac{a}{b^i}\right)^k + cb^{mi} \\
&= a^m + cb^{mi} \sum_{k=0}^{m-2} \left(\tfrac{a}{b^i}\right)\left(\tfrac{a}{b^i}\right)^k + cb^{mi} \\
&= a^m + cb^{mi} \left[\sum_{k=0}^{m-2} \left(\tfrac{a}{b^i}\right)^{k+1} + 1\right]
\end{aligned}$$

$$\text{But } \sum_{k=0}^{m-2} \left(\frac{a}{b^i}\right)^{k+1} + 1 = \sum_{k=0}^{m-1} \left(\frac{a}{b^i}\right)^k$$

$$\text{Therefore } T(b^m) = a^m \ + \ cb^{mi} \sum_{k=0}^{m-1} \left(\frac{a}{b^i}\right)^k$$

which is what we wanted to show. And therefore, the closed form is true for **any** value of $j > 0$, i.e.

$$T(b^j) = a^j + cb^{ji} \sum_{k=0}^{j-1} \left(\frac{a}{b^i}\right)^k$$

We can still write the above in a more precise form by calculating the above summation

$$\sum_{k=0}^{j-1} \left(\frac{a}{b^i}\right)^k$$

which depends on the relationship between a and b^i.

- **Case 1:** $a = b^i$**:**

$$a = b^i \Rightarrow \sum_{k=0}^{j-1} \left(\frac{a}{b^i}\right)^k = \sum_{k=0}^{j-1} 1 = j$$

$$\Rightarrow T(b^j) = a^j + cjb^{ji} \Rightarrow T(b^j) = (b^i)^j + cjb^{ji}$$

$$\Rightarrow T(b^j) = b^{ji}(1 + cj)$$

- **Case 2:** $a > b^i$**:**

$$\textit{Recall: } \sum_{k=0}^{j-1} n^k = \frac{n^j - 1}{n - 1} \text{ if } n \neq 1$$

$$\Rightarrow T(b^j) = a^j + cb^{ji} \left[\frac{\left(\frac{a}{b^i}\right)^j - 1}{\left(\frac{a}{b^i}\right) - 1}\right]$$

- **Case 3:** $a < b^i$

$$T(b^j) = a^j + cb^{ji} \left[\frac{\left(\frac{a}{b^i}\right)^j - 1}{\left(\frac{a}{b^i}\right) - 1}\right]$$

$$\text{but since } \left(\frac{a}{b^i}\right)^j - 1 < 0 \text{ and } \frac{a}{b^i} - 1 < 0 \text{ rewrite it:}$$

$$T(b^j) = a^j + cb^{ji} \left[\frac{1 - \left(\frac{a}{b^i}\right)^j}{1 - \left(\frac{a}{b^i}\right)}\right]$$

Since $n = b^j$, let us calculate the growth rates with respect to n.
¿From Mathematics we know:

- $n = b^j \Rightarrow \log_b n = \log_b b^j \Rightarrow \log_b n = j$
- $a^{\log_b n} = \left(\underbrace{b^{\log_b a}}_{\text{a}}\right)^{\log_b n} = b^{(\log_b a)(\log_b n)} = \left(\underbrace{b^{\log_b n}}_{\text{n}}\right)^{\log_b a} = n^{\log_b a}$
- $\left(\frac{a}{b^i}\right)^{\log_b n} = \frac{a^{\log_b n}}{b^{i \log_b n}} = \frac{n^{\log_b a}}{b^{\log_b n^i}} = \frac{n^{\log_b a}}{n^i} = n^{\log_b a - i}$
- **Case 1:** $a = b^i$**:**
  - $a = b^i \Rightarrow \log_b a = \log_b b^i \Rightarrow \log_b a = i$
  - $T(b^j) = (b^j)^i(1 + cj)$

$$\Rightarrow T(n) = n^i(1 + c \log_b n)$$

$$\text{Therefore } T(n) \text{ is } \Theta(n^i \log_b n)$$

- **Case 2:** $a > b^i$**:**
  - $a > b^i \Rightarrow i < \log_b a$
  - $b^j = n \Rightarrow j = \log_b n$
  - $T(b^j) = a^j + cb^{ji} \left[\frac{\left(\frac{a}{b^i}\right)^j - 1}{\frac{a}{b^i} - 1}\right]$

$$\Rightarrow T(n) = n^{\log_b a} + cn^i \left[\frac{n^{\log_b a - i} - 1}{\frac{a}{b^i} - 1}\right] = n^{\log_b a} + \frac{cn^{\log_b a} - cn^i}{\frac{a}{b^i} - 1}$$

$$T(n) = n^{\log_b a}\left[1 + \frac{c}{\frac{a}{b^i} - 1}\right] - \frac{cn^i}{\frac{a}{b^i} - 1}$$

Thus the leading function of $T(n)$ is $c_1 n^{\log_b a}$ or $c_2 n^i$.
But $a > b^i \Rightarrow \log_b a > i$ and therefore

$$T(n) \text{ is } \Theta(n^{\log_b a})$$

- **Case 3:** $a < b^i$**:**
  - $a < b^i \Rightarrow i > \log_b a$
  - $T(b^j) = a^j + cb^{ji}\left[\frac{1-\left(\frac{a}{b^i}\right)^j}{1-\frac{a}{b^i}}\right]$

$$\begin{aligned}
\Rightarrow T(n) &= n^{\log_b a} + cn^i\left[\frac{1-n^{\log_b a - i}}{1-\frac{a}{b^i}}\right] \quad \text{(see Case 2)} \\
&= n^{\log_b a} + c\left[\frac{n^i - n^{\log_b a}}{1-\frac{a}{b^i}}\right] \\
&= n^{\log_b a}\left[1 - \frac{c}{1-\frac{a}{b^i}}\right] + n^i\left[\frac{c}{1-\frac{a}{b^i}}\right]
\end{aligned}$$

And with $i > \log_b a$ we have

$$T(n) \text{ is } \Theta(n^i)$$

We summarize the results in the next section.

2.2.3 Summary

The solution of the following recurrence relation

$$\begin{array}{lcll}
T(1) & = & 1 & \\
T(n) & = & aT(\frac{n}{b}) + cn^i & n > 1
\end{array}$$

depends on the relationship between a and b^i:

Case	Growth Rate
$a < b^i$	$\Theta(n^i)$
$a = b^i$	$\Theta(n^i \log_b n)$
$a > b^i$	$\Theta(n^{\log_b a})$

And the closed form is:

$$T(b^j) = a^j + cb^{ji}\sum_{k=0}^{j-1}\left(\frac{a}{b^i}\right)^k$$

2.2.4 Rates of growth of some divide&conquer algorithms

- **Binary Search:**
  Worst case analysis: $W(n) = 1 + W(\lfloor \frac{n}{2} \rfloor)$
  With $n = 2^j \Rightarrow W(n) = 1 + W(\frac{n}{2})$

$$\begin{array}{lrcl} \text{i.e.} & W(2^j) & = & 1 + W(\frac{2^j}{2}) \\ \text{and} & W(1) & = & 1 \end{array}$$

  So $a = 1$, $b = 2$ $i = 0$ and $c = 1$.
  $a = 1 = 2^0 = b^i$, therefore the answer is given by Case 1:

$$T(b^j) = b^{ji}(1 + cj)$$

$$\text{So } W(n) = n^0(1 + 1\log_2 n) = 1 + \log_2 n$$

$$\Rightarrow \Theta(\log_2 n)$$

- Max-Min **Problem:**
  $C(n)$ Total number of comparisons required to find the maximum and minimum values for an array A with n elements

$$C(1) = 0,\, C(2) = 1$$

$$C(n) = 2C\left(\frac{n}{2}\right) + 2 \quad \text{if } n > 2 \text{ is even.}$$

$$C(n) = C\left(\frac{n+1}{2}\right) + C\left(\frac{n-1}{2}\right) + 2 \quad \text{if } n > 2 \text{ is odd.}$$

  If $n = 2^j$ i.e.:

$$\begin{array}{rcl} C(2^j) & = & 2C\left(\frac{2^j}{2}\right) + 2 \\ C(1) & = & 0 \end{array}$$

  So $a = 2$, $b = 2$, $i = 0$ and $c = 2 \Rightarrow a > b^i$.
  Therefore the answer is given by Case 2:

$$C(n) \text{ is } \Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n)$$

- Mergesort **Problem:**
  The number of comparisons used by Mergesort can be described by the following recurrence relation:

$$C(n) = C\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + C\left(\left\lceil \frac{n}{2} \right\rceil\right) + n$$

  If $n = 2^j$

$$\Rightarrow C(n) = 2C\left(\frac{n}{2}\right) + cn$$

  i.e. $a = 2$, $b = 2$ and $i = 1 \Rightarrow a = b^i$ so the answer is given by Case 1:

$$C(n) \text{ is } \Theta(n^i \log_b n) = \Theta(n^1 \log_2 n)$$

$$\Rightarrow \Theta(n \log_2 n)$$

  Note: Exact solution:

$$C(n) = n\lfloor \log_2 n \rfloor + 2n - 2^{\lfloor \log_2 n \rfloor + 1}$$

In the next section we study Strassen's matrix multiplication algorithm.

2.3 Strassen's matrix multiplication [CLR90, Section 31.2]

Given two $n \times n$ matrices A and B, the straightforward algorithm to compute the product matrix C will have a complexity of $\Theta(n^3)$.

Suppose n is even. Partition A and B into $\frac{n}{2} \times \frac{n}{2}$ submatrices:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \qquad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

The product matrix, $C = AB$, can be written as:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

Consequently:

$$\begin{aligned} C_{11} &= A_{11}B_{11} + A_{12}B_{21} \\ C_{12} &= A_{11}B_{12} + A_{12}B_{22} \\ C_{21} &= A_{21}B_{11} + A_{22}B_{21} \\ C_{22} &= A_{21}B_{12} + A_{22}B_{22} \end{aligned}$$

Each of the 4 equations comprises of 2 multiplications of $\frac{n}{2} \times \frac{n}{2}$ matrices and the addition of their $\frac{n}{2} \times \frac{n}{2}$ products.

So a divide and conquer algorithm based on the above strategy will have as running time function (we assume here that n is a power of 2):

$$\begin{aligned} T(n) &= 8T\left(\tfrac{n}{2}\right) + cn^2 \quad n > 1 \\ T(n) &= 1 \end{aligned}$$

Here $a = 8$, $b = 2$ and $i = 2$, so $a > b^i$ and $T(n) = \Theta(n^{\log_2 8})$ (see Case 2). So $T(n)$ is $\Theta(n^3)$ which is not an improvement over the straightforward algorithm.

Strassen's method:

Strassen reduced the number of subproblems from 8 to 7 by using the following idea:

Define the following:

$$\begin{aligned} P &= (A_{11} + A_{22})(B_{11} + B_{22}) \\ Q &= (A_{21} + A_{22})B_{11} \\ R &= A_{11}(B_{12} - B_{22}) \\ S &= A_{22}(B_{21} - B_{11}) \\ T &= (A_{11} + A_{12})B_{22} \\ U &= (A_{21} - A_{11})(B_{11} + B_{12}) \\ V &= (A_{12} - A_{22})(B_{21} + B_{22}) \end{aligned}$$

We can easily check that:

$$\begin{aligned} C_{11} &= P + S - T + V \\ C_{12} &= R + T \\ C_{21} &= Q + S \\ C_{22} &= P + R - Q + U \end{aligned}$$

So a divide and conquer algorithm based on Strassen's strategy will have a running time function (assuming again that n is a power of 2):

$$\begin{aligned} T(n) &= 7T(\tfrac{n}{2}) + dn^2 \quad n > 1 \\ T(1) &= 1 \end{aligned}$$

Here $a = 7$, $b = 2$ and $i = 2$, so $T(n)$ is $\Theta(n^{\log_2 7}) = \Theta(n^{2.81})$.
But is Strassen's algorithm really better than the straightforward method ?

It depends on the value of n and whether the matrix is dense (few zero entries) or not. For large values of n (greater than 50), Strassen's algorithm is preferable. Why ?

We need more work to break down the problem into 7 subproblems: Multiplications and sums (or differences) of $\frac{n}{2} \times \frac{n}{2}$ matrices, instead of the 4 we have with the first divide and conquer algorithm.

By using more advanced techniques, we can multiply two $n \times n$ matrices in $\Theta(n^{2.376})$.

In the next section, we study another algorithm that is based on the divide and conquer technique.

2.4 Fast Fourier Transform [CLR90, Chapter 32]

1. **Multiplication of polynomials**
   If we have two polynomials, given in coefficient form:

$$A(x) = \sum_{j=0}^{n-1} a_j x j \qquad\qquad B(x) = \sum_{j=0}^{n-1} b_j x^j$$

   then we can write the product

$$C(x) = A(x)B(x) = \sum_{j=0}^{2n-2} c_j x^j \text{ where } c_j = \sum_{k=0}^{j} a_k b_{j-k}$$

   EXAMPLE:

$$A(x) = x^2 + 3x - 1\,; \qquad B(x) = x^2 - 2$$

$$\text{i.e.} \quad \begin{array}{lcrcrcrcrcr} A(x) & = & -1 & + & 3x & + & x^2 & + & 0x^3 & + & 0x^4 \\ B(x) & = & -2 & + & 0x & + & x^2 & + & 0x^3 & + & 0x^4 \end{array}$$

   and we get the following (*Note*: $n = 3$)

$$\begin{array}{lclclclclclcl} c_0 & = & a_0b_0 & = & 2 \\ c_1 & = & a_0b_1 & + & a_1b_0 & = & -6 \\ c_2 & = & a_0b_2 & + & a_1b_1 & + & a_2b_0 & = & -3 \\ c_3 & = & a_0b_3 & + & a_1b_2 & + & a_2b_1 & + & a_3b_0 & = & 3 \\ c_4 & = & a_0b_4 & + & a_1b_3 & + & a_2b_2 & + & a_3b_1 & + & a_4b_0 & = & 1 \end{array}$$

   So $C(x) = 2 - 6x - 3x^2 + 3x^3 + x^4$.
   Clearly, computing $C(x) = A(x)B(x)$ has a growth rate of $\Theta(n^2)$.

2. **Point-value representation**
   A polynomial $A(x)$ of degree $n-1$ can be represented by n point-value pairs:

$$\{(x_0, y_0),\, (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})\}$$

   where:

$$\text{a)} \quad y_k = A(x_k) \qquad\qquad \text{b) all } x_k \text{ are distinct.}$$

   A polynomial has many different point-value representations, since any set of n distinct points x_0, x_1 … x_{n-1} can be used as a basis for the representation.

   EXAMPLE:

$$A(x) = -1 + 3x + x^2 \qquad\qquad B(x) = -2 + x^2$$

One possible point-value representation of $A(x)$ and $B(x)$:

$$\begin{array}{rcl} A(x) & = & \{(-1,-3),(1,3),(2,9)\} \\ B(x) & = & \{(-1,-1),(1,-1),(2,2)\} \end{array}$$

(a) Coefficient form $\longrightarrow$ Point-value representation
To compute a point-value representation for a given polynomial in coefficient form, choose n distinct points $x_0, \ldots, x_{n-1}$ and evaluate $A(x_k)$ for $k = 0, 1, \ldots, n-1$ by using Horner's method:

$$A(x_k) = a_0 + x_k(a_1 + x_k(a_2 + \cdots + x_k(a_{n-2} + x_k a_{n-1}))\cdots)$$

Since it takes $\Theta(n)$ operations to evaluate $A(x_k)$ for each x_k, it takes $\Theta(n^2)$ to evaluate all n points. This operation is known as **evaluation**.

(b) Point-value representation $\longrightarrow$ Coefficient form (a.k.a. **interpolation**).
Interpolation is the process of determining the coefficient form of a polynomial from a point-value representation. Lagrange's formula is often used for interpolation:

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod^{j \neq k} (x - x_j)}{\prod^{j \neq k} (x_k - x_j)}$$

EXAMPLE:

$$A(x) = \{(-1,-3),(1,3),(2,9)\}$$

$$A(x) = \sum_{k=0}^{n-1} y_k \frac{\prod^{j \neq k} (x - x_j)}{\prod^{j \neq k} (x_k - x_j)}$$

$$\Rightarrow A(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} +$$

$$+ y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$= -\frac{1}{2}(x-1)(x-2) - \frac{3}{2}(x+1)(x-2) + 3(x+1)(x-1)$$

$$\Rightarrow A(x) = x^2 + 3x - 1$$

Result: Interpolation is a well-defined operation.
Given any set $\{(x_0,y_0),(x_1,y_1),\ldots,(x_{n-1},y_{n-1})\}$ of n point-value pairs, there is a unique polynomial $A(x)$ of degree n such that:

$$y_k = A(x_k) \text{ for } k = 0, 1, \ldots\ n-1$$

The proof uses the Vandermonde matrix.

3. **Operations in point-value representation**
$A(x)$ and $B(x)$ are two polynomials in point-value representation where the first component of the pairs of the two polynomials are the same:

$$A(x) = \{(x_0,y_0),(x_1,y_1),\ldots,(x_{n-1},y_{n-1})\}$$

$$B(x) = \{(x_0,y'_0),(x_1,y'_1),\ldots,(x_{n-1},y'_{n-1})\}$$

$$\text{Then: } C(x) = A(x) + B(x)$$

$$\Rightarrow C(x) = \{(x_0,y_0+y'_0),(x_1,y_1+y'_1),\ldots,(x_{n-1},y_{n-1}+y'_{n-1})\}$$

For multiplication, extend the point-value representations:

$$A(x) = \{(x_0, y_0) \cdots (x_{n-1}, y_{n-1}), (x_n, y_n), \ldots (x_{2n-1}, y_{2n-1})\}$$
$$B(x) = \{(x_0, y_0') \ldots (x_{n-1}, y_{n-1}'), (x_n, y_n'), \ldots (x_{2n-1}, y_{2n-1}')\}$$

Then a point-value representation for

$$C(x) = A(x)B(x) \text{ is:}$$

$$C(x) = \{(x_0, y_0y_0'), (x_1, y_1y_1') \ldots (x_{2n-1}, y_{2n-1}y_{2n-1}')\}.$$

So now we have a $\Theta(n)$ procedure to multiply two polynomials in point-value representation.

EXAMPLE:
Extend

$$\begin{aligned} &A(x) = \{(-1,-3),(1,3),(2,9)\} \\ \text{and}\quad &B(x) = \{(-1,-1),(1,-1),(2,2)\} \end{aligned}$$

to obtain:

$$A(x) = \{(-1,-3),(1,3),(2,9),(-2,-3),(0,-1)\}$$

and

$$B(x) = \{(-1,-1),(1,-1),(2,2),(-2,2),(0,-2)\}$$

Recall: $A(x) = x^2 + 3x - 1$ and $B(x) = x^2 - 2$.
Compute $C(x) = A(x)B(x)$ from the point-value representations of $A(x)$ and $B(x)$:

$$C(x) = \{(-1,3),(1,-3),(2,18),(-2,-6),(0,2)\}$$

We can write $C(x)$ in coefficient form by using Lagrange's formula; but it is easier to check and see that $C(x)$ here is indeed $C(x) = x^4 + 3x^3 - 3x^2 - 6x + 2$, since $C(-1) = 3$, $C(1) = -3$, $C(2) = 18$, $C(-2) = -6$ and $C(0) = 2$. □

So we have:
Conversion between coefficient representation and point-value representation requires:

(a) generating $2n$ points
(b) evaluating $y_k = A(x_k)$ is an $\Theta(n)$ operation using Horner's rule.

So the conversion requires $\Theta(n^2)$ operations.

We can do better by choosing the evaluation points more carefully.

4. **Complex roots of unity**
A complex n^{th} root of unity is a complex number ω such that

$$\omega^n = 1\,.$$

There are exactly n such roots. These are

$$e^{\frac{2\pi ik}{n}} \text{ for } k = 0, 1, 2, \ldots, n-1\,.$$

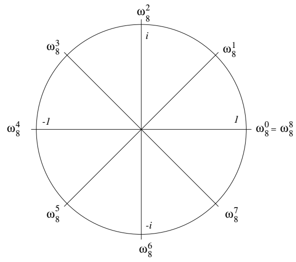
We use the definition of the exponential of a complex number

$$e^{iu} = \cos(u) + i\sin(u)$$

to interpret $e^{\frac{2\pi ik}{n}}$.

$$\text{So } e^{\frac{2\pi ik}{n}} = \cos\left(\frac{2\pi k}{n}\right) + i\sin\left(\frac{2\pi k}{n}\right) \text{ for } k = 0, 1, 2, \ldots, n-1\,.$$

EXAMPLE: $n = 8$



The 8 complex roots of unity $\omega_8^0, \omega_8^1, \ldots, \omega_8^7$ are equally spaced around the circle of unit radius at the origin of the complex plane. □

The principle n^{th} root of unity is

$$\omega_n = e^{\frac{2\pi i}{n}}.$$

The other roots: $\omega_n^2, \omega_n^3, \omega_n^4, \ldots, \omega_n^{n-1}$ are powers of the principle n^{th} root of unity.

Remark: The n complex n^{th} roots of unity, $G_n = \{w_0^0, w_n^1, w_n^2, \ldots, w_n^{n-1}\}$, form a group under multiplication $\circ$.

- Closure

$$\forall\ w_n^j \in G_n \wedge w_n^k \in G_n :\ w_n^j \circ w_n^k \in G_n$$

  Note that

$$w_n^j \circ w_n^k = w_n^{j+k} = w_n^{(j+k) \bmod n}$$

$\circ$	w_4^0	w_4^1	w_4^2	w_4^3
w_4^0	1	w_4^1	w_4^2	w_4^3
w_4^1	w_4^1	w_4^2	w_4^3	1
w_4^2	w_4^2	w_4^3	1	w_4^1
w_4^3	w_4^3	1	w_4^1	w_4^2

- Identity element: $w_n^0 = 1$

$$\forall\ w_n^j \in G_n : w_n^j \circ w_n^0 = w_n^0 \circ w_n^j = w_n^j$$

- Associativity:

$$\forall\ w_n^j, w_n^k, w_n^l \in G_n : \left(w_n^j \circ w_n^k\right) \circ w_n^l = w_n^j \circ \left(w_n^k \circ w_n^l\right)$$

- Inverses:

$$\forall\ w_n^j \in G_n, \exists! w_n^k = w_n^{n-j} : w_n^j \circ w_n^{n-j} = w_n^{n-j} \circ w_n^j = w_n^n = w_n^0 = 1$$

5. **Properties of ω**

   (a) Cancellation Lemma:
   For any $n \geq 0$, $k \geq 0$, $d \geq 0$ we have $\omega_{dn}^{dk} = \omega_n^k$.
   Proof:

$$\begin{aligned} \omega_{dn}^{dk} &= \left(e^{2\pi i/dn}\right)^{dk} \\ &= \left(e^{2\pi i/n}\right)^{k} \\ &= \omega_n^k \end{aligned}$$

   (b) *For any even integer $n > 0$, $\omega_n^{n/2} = \omega_2 = -1$.* In the example $\omega_8^4 = -1$ and $\omega_2 = e^{\pi i} = \cos\pi + i\sin\pi = -1$.

(c) Halving Lemma:

If $n > 2$ and even, then the squares of the n complex n^{th} roots of unity are the $\frac{n}{2}$ complex $\left(\frac{n}{2}\right)^{th}$ roots of unity (i.e. they evaluate to 1 when raised to $\left(\frac{n}{2}\right)^{th}$ power).

Proof:

By the cancellation lemma, we know that

$$\left(w_n^k\right) = w_{2\cdot\left(\frac{n}{2}\right)}^{2k} = w_{\frac{n}{2}}^k \quad \forall\ k > 0$$

And for $\omega_n^{k+n/2}$ we get:

$$\begin{aligned} \left(\omega_n^{k+n/2}\right)^2 &= \omega_n^{2k+n} \\ &= \omega_n^{2k}\omega_n^n \\ &= \omega_n^{2k} \\ &= \left(\omega_n^k\right)^2 \\ &= \omega_{n/2}^k \end{aligned}$$

$\square$

Example: w_8^3 is an 8^{th} root of unity; $\left(w_8^3\right)^2 = w_{2\cdot 4}^{2\cdot 3} = w_4^3$ is a 4^{th} root of unity.

Note that in the above proof, we have also shown that by squaring the complex n^{th} roots of unity, we obtain each $\left(\frac{n}{2}\right)^{th}$ root of unity exactly twice.

For Example: The 8 complex 8^{th} roots of unity are:

$$w_8^0,\ w_8^1,\ w_8^2,\ w_8^3,\ w_8^4,\ w_8^5,\ w_8^6,\ w_8^7$$

By squaring each one, we have:

$$w_8^0,\ w_8^2,\ w_8^4,\ w_8^6,\ w_8^8 = w_8^0,\ w_8^{10} = w_8^2,\ w_8^{12} = w_8^4,\ w_8^{14} = w_8^6$$

which is equivalent (by the cancellation lemma) to:

$$w_4^0,\ w_4^1,\ w_4^2,\ w_4^3,\ w_4^0,\ w_4^1,\ w_4^2,\ w_4^3$$

Thus we see that by squaring all complex 8^{th} roots of unity, each $\left(\frac{8}{2}\right)^{th}$ root of unity is obtained exactly twice.

In other words, w_n^k and $w_n^{k+n/2}$ have the same square (and will therefore give two copies of an $\frac{n}{2}^{th}$ root of unity).

Note that the proof of this theorem is just part of the above proof. In other words:

Proof:

$$\begin{aligned} \left(\omega_n^{k+n/2}\right)^2 &= \omega_n^{2k+n} \\ &= \omega_n^{2k}\omega_n^n \\ &= \omega_n^{2k}(1) \\ &= \left(\omega_n^k\right)^2 \end{aligned}$$

An alternative proof:

$$w_n^{n/2} = -1 \ \Rightarrow\ w_n^{k+n/2} = w_n^{n/2}w_n^k = -w_n^k$$

and so $\left(w_n^{k+n/2}\right)^2 = \left(w_n^k\right)^2$ $\square$

(d) Summation Lemma:
For any integer $n \geq 1$ and nonnegative integer k not divisible by n, we have:

$$\sum_{j=0}^{n-1} \left(w_n^k\right)^j = 0$$

Proof: Recall

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1} \quad \text{for } x \neq 1 \text{ geometric series}$$

$$\begin{aligned}
\sum_{j=0}^{n-1} \left(w_n^k\right)^j &= \frac{\left(w_n^k\right)^n - 1}{w_n^k - 1} = \\
&= \frac{(w_n^n)^k - 1}{w_n^k - 1} = \\
&= \frac{1^k - 1}{w_n^k - 1} = \\
&= 0
\end{aligned}$$

Note that $w_n^k \neq 1$ since k is not divisible by n. □

6. **The Fast Fourier Transform**
Rather than dividing the polynomial in the middle (as is the case with other divide and conquer algorithms) we'll divide it into two parts by putting alternate terms in each part. To evaluate

$$A(x) = \sum_{j=0}^{n-1} a_j x^j \,, \; n \text{ is even}$$

we divide it into two parts:

$$A^{even}(x) = a_0 + a_2 x + a_4 x^2 + \ldots + a_{n-2} x^{\frac{n}{2}-1}$$

and

$$A^{odd}(x) = a_1 + a_3 x + a_5 x^2 + \ldots + a_{n-1} x^{\frac{n}{2}-1} \,.$$

$A^{even}(x)$ contains all even-index coefficients of $A(x)$. $A^{odd}(x)$ contains the odd-index coefficients. We have

$$A(x) = A^{even}(x^2) + x A^{odd}(x^2) \,.$$

The n^{th} roots of unity are convenient for this decomposition because as we saw above, squaring a root of unity gives us another root of unity. So evaluating $A(x)$ at $\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$ reduces to:

(a) evaluating $A^{even}(x)$ and $A^{odd}(x)$ at $\left(\omega_n^0\right)^2, \left(\omega_n^1\right)^2, \left(\omega_n^2\right)^2, \cdots, \left(\omega_n^{n-1}\right)^2$. But these are the complex $\left(\frac{n}{2}\right)^{th}$ roots of unity. By the halving Lemma we know that there are only $\frac{n}{2}$ roots.

(b) Combine the results according to: $A(x) = A^{even}(x^2) + x A^{odd}(x^2)$.

7. **Recursive Fast Fourier Transform**
Assume that n is a power of 2.

```
RecurFFT(a, n)                        // a is the list of n coefficients
begin
    if n = 1 then return a
    ω_n = e^(2πi/n)
    ω = 1
    A^even = (a_0, a_2, a_4 ··· , a_{n-2})
    A^odd = (a_1, a_3, a_5 ··· , a_{n-1})
    y^even = RecurFFT(A^even, n/2)
    y^odd = RecurFFT(A^odd, n/2)
    for k = 0 to n/2 - 1
        y_k = y_k^even + ω y_k^odd
        y_{k+n/2} = y_k^even - ω y_k^odd
        ω = ω ω_n
    return y
end
```

8. Example:
Evaluate

$$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_7 x^7$$

at the 8 complex 8^{th} roots of unity

$$w_8^0,\ w_8^1,\ w_8^2,\ \ldots, w_8^7 \quad (*)$$

Recall: Since $w_8^4 = -1$ (property (b)), evaluating $A(x)$ at points $(*)$ is the same as evaluating $A(x)$ at:

$$w_8^0,\ w_8^1,\ w_8^2,\ w_8^3,\ -w_8^0,\ -w_8^1,\ -w_8^2,\ -w_8^3$$

Squaring each term gives two copies of the 4 complex roots of the 4^{th} roots of unity:

$$w_4^0,\ w_4^1,\ w_4^2,\ w_4^3,\ w_4^0,\ w_4^1,\ w_4^2,\ w_4^3$$

So by using:

$$A(x) = A^{even}(x^2) + xA^{odd}(x^2)$$

we can recombine the results of computing A^{even} and A^{odd} at the 4^{th} roots of unity. More precisely:

(a) Evaluate $A^{even}(x)$ and $A^{odd}(x)$ at the 4^{th} roots of unity.

(b) Substitute each of the 8^{th} roots of unity for x in

$$A(x) = A^{even}(x^2) + xA^{odd}(x^2)$$

$$\begin{array}{rcccl}
A(w_8^0) & = & A^{even}(\overbrace{w_4^0}^{=(w_8^0)^2}) & + & w_8^0 A^{odd}(\overbrace{w_4^0}^{=(w_8^0)^2}) \\
A(w_8^1) & = & A^{even}(w_4^1) & + & w_8^1 A^{odd}(w_4^1) \\
A(w_8^2) & = & A^{even}(w_4^2) & + & w_8^2 A^{odd}(w_4^2) \\
A(w_8^3) & = & A^{even}(w_4^3) & + & w_8^3 A^{odd}(w_4^3) \\
A(w_8^4) & = & A^{even}(w_4^0) & - & w_8^0 A^{odd}(w_4^0) \\
A(w_8^5) & = & A^{even}(w_4^1) & - & w_8^1 A^{odd}(w_4^1) \\
A(w_8^6) & = & A^{even}(w_4^2) & - & w_8^2 A^{odd}(w_4^2) \\
A(w_8^7) & = & A^{even}(w_4^3) & - & w_8^3 A^{odd}(w_4^3)
\end{array}$$

For instance:

$$
\begin{aligned}
A(w_8^0) &= a_0 + a_1 w_8^0 + a_2(w_8^0)^2 + a_3(w_8^0)^3 + \\
&\quad a_4(w_8^0)^4 + a_5(w_8^0)^5 + a_6(w_8^0)^6 + a_7(w_8^0)^7 \\
A(w_8^0) &= a_0 + a_2(w_8^0)^2 + a_4(w_8^0)^4 + a_6(w_8^0)^6 + \\
&\quad w_8^0[a_1 + a_3(w_8^0)^2 + a_5(w_8^0)^4 + a_7(w_8^0)^6] \\
A(w_8^0) &= a_0 + a_2 w_4^0 + a_4(w_4^0)^2 + a_6(w_4^0)^3 + \\
&\quad w_8^0[a_1 + a_3 w_4^0 + a_5(w_4^0)^2 + a_7(w_4^0)^3] \\
\Rightarrow\ A(w_8^0) &= A^{even}(w_4^0) + w_8^0 A^{odd}(w_4^0)
\end{aligned}
$$

and also,

$$
\begin{aligned}
A(w_8^4) &= a_0 + a_1 w_8^4 + a_2(w_8^4)^2 + a_3(w_8^4)^3 + \\
&\quad a_4(w_8^4)^4 + a_5(w_8^4)^5 + a_6(w_8^4)^6 + a_7(w_8^4)^7 \\
A(w_8^4) &= a_0 + a_1(-w_8^0) + a_2(-w_8^0)^2 + a_3(-w_8^0)^3 + \\
&\quad a_4(-w_8^0)^4 + a_5(-w_8^0)^5 + a_6(-w_8^0)^6 + a_7(-w_8^0)^7 \\
&= a_0 + a_2(w_8^0)^2 + a_4(w_8^0)^4 + a_6(w_8^0)^6 - w_8^0[a_1 + \\
&\quad a_3(w_8^0)^2 + a_5(w_8^0)^4 + a_7(w_8^0)^6] \\
A(w_8^4) &= a_0 + a_2 w_4^0 + a_4(w_4^0)^2 + a_6(w_4^0)^3 - \\
&\quad w_8^0[a_1 + a_3 w_4^0 + a_5(w_4^0)^2 + a_7(w_4^0)^3] \\
\Rightarrow\ A(w_8^0 &= A^{even}(w_4^0) - w_8^0 A^{odd}(w_4^0)
\end{aligned}
$$

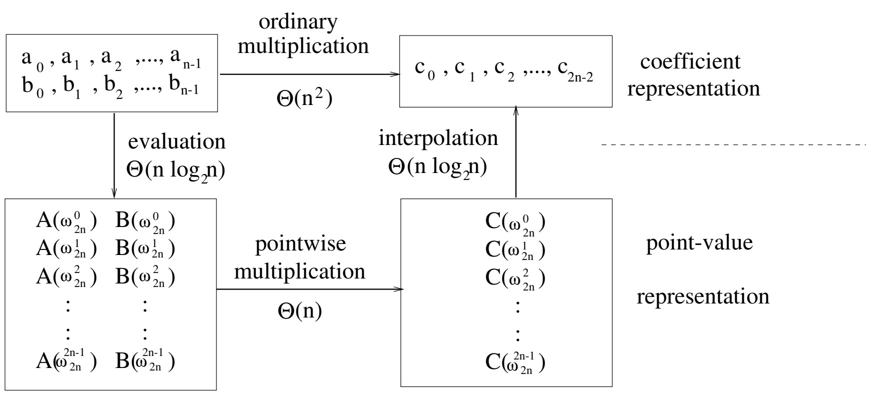
In general, to evaluate $A(x)$ at the n^{th} roots of unity, recursively evaluate $A^{even}(x)$ and $A^{odd}(x)$ at the $\frac{n}{2}^{th}$ roots of unity and perform the n substitutions as was done with $n = 8$.

9. **Analysis of the algorithm**
The running time function of RecurFFT is given by

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)\,.$$

Here $a = 2$, $b = 2$ and $i = 1$, so $a = b^i$. And therefore T(n) is $\Theta(n \log_2 n)$.

The whole process of efficiently multiplying two polynomials is summarized in the following graph:



In the next section we study another family of recurrence relations, that are not generally encountered when studying algorithms based on the divide and conquer approach.

2.5 Other recurrence relations

Recurrence relations that do not necessarily come from divide and conquer algorithms.

BUBBLE SORT:

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

The recurrence relation can be solved by:

(a) **Substitution**: Then use induction to prove that the closed form is correct

(b) using **homogeneous linear equation technique**

(c) using **generating functions**.

2.5.1 Linear recurrence relations with constant coefficients [Liu68, Chapter 7],[Liu85, Chapter 10]

$$c_0T(n) + c_1T(n-1) + c_2T(n) + \ldots + c_kT(n-k) = f(n)$$

is a linear recurrence relation with constant coefficients of k^{th} order. Here c_i's are constants, $c_0 \neq 0\,,\ c_k \neq 0$.

Remark: If k consecutive values are know, the other values can be determined (i.e. $T(n)$ will be uniquely defined).

EXAMPLE:

$$3T(n) - 5T(n-1) + 2T(n-2) = n^2 + 5$$

is a $2nd$ order linear recurrence relation with constant coefficients. If $T(3)$=3 and $T(4) = 6$, then:

$$3T(4) - 5T(3) + 2T(2) = 4^2 + 5$$

$$\Rightarrow T(2) = 9$$

$$3T(5) - 5T(4) + 2T(3) = 5^2 + 5 \Rightarrow T(5) = 18 \text{ and so on...}$$

k non-consecutive values might not constitute an appropriate set of boundary conditions (problem dependent).

The total solution of a linear recurrence relation with constant coefficients is given by:

$$T(n) = T_h(n) + T_p(n)\,,$$

where

a.) $T_h(n)$ is the homogeneous solution and satisfies the linear recurrence relation when the right-hand side of the equation is set to 0 (i.e. $f(n) = 0$).

b.) $T_p(n)$ is the particular solution and satisfies the linear recurrence relation with $f(n)$ on the right-hand side.

A homogeneous solution of a linear recurrence relation is of the form $A\alpha^n$, where α is a characteristic root and A is a constant determined by the boundary conditions.

Note that by substituting $A\alpha^n$ for $T(n)$, we get:

$$c_0A\alpha^n + c_1A\alpha^{n-1} + c_2A\alpha^{n-2} + \cdots + c_kA\alpha^{n-k} = 0$$

$$\Rightarrow\ A\alpha^{n-k}(c_0\alpha^k + c_1\alpha^{k-1} + \cdots + c_k) = 0$$

$$\Leftrightarrow\ c_0\alpha^k + c_1\alpha^{k-1} + \cdots + c_k = 0$$

which is called the characteristic equation of the recurrence relation. The characteristic equation of the k^{th} degree has k characteristic roots.

1. If the k characteristic roots are distinct then:

$$T_h(n) = A_1\alpha_1^n + A_2\alpha 2^n + \cdots + A_k\alpha_k^n$$

where $\alpha_1, \alpha_2, \ldots, \alpha_k$ are the roots of the characteristic equation and $A_1, A_2, \ldots, A_k$ are constants determined by the boundary conditions.

2. If the k characteristic roots are not distinct, then some of the roots are multiple roots. If α_1 is a root of multiplicity m, then:

$$(A_1n^{m-1} + A_2n^{m-2} + \cdots + A_{m-1}n + A_m)\alpha^n$$

is a homogeneous solution.

EXAMPLES:

1. Solve the following recurrence relation:

$$T(n) = T(n-1) + T(n-2)$$
$$\left.\begin{array}{ll} T(0) & = 0 \\ T(1) & = 1 \end{array}\right\} \text{boundary conditions}$$

(i) Find $T_h(n)$, so solve $T(n) - T(n-1) - T(n-2) = 0$. Note that $f(n) = 0$ and so there is no particular solution here i.e. $T_p(n) = 0$.

$$T(n) = A\alpha^n \;\Rightarrow A\alpha^n - A\alpha^{n-1} - A\alpha^{n-2} = 0$$
$$\Rightarrow\; A\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$$

$\alpha^2 - \alpha - 1 = 0$ is the characteristic equation.

$$\Rightarrow\; \alpha_1 = \frac{1+\sqrt{5}}{2} \text{ and } \alpha_2 = \frac{1-\sqrt{5}}{2} \text{ are the two distinct roots.}$$

Therefore

$$T_h(n) = A_1\left(\frac{1-\sqrt{5}}{2}\right)^n + A_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

is the homogeneous solution.

(ii) to calculate A_1 and A_2 use the boundary conditions:

$$T_h(0) = A_1\left(\frac{1+\sqrt{5}}{2}\right)^0 + A_2\left(\frac{1-\sqrt{5}}{2}\right)^0$$
$$\Rightarrow\; A_1 + A_2 = 1 \quad (\text{since } T(0) = 1 \text{ given})$$
$$T_h(1) = A_1\left(\frac{1+\sqrt{5}}{2}\right) + A_2\left(\frac{1-\sqrt{5}}{2}\right)$$
$$\Rightarrow\; \frac{(A_1 + A_2) + \sqrt{5}(A_1 - A_2)}{2} = 1 \quad (\text{since } T(1) = 1 \text{ given})$$
$$\Rightarrow\; A_1 - A_2 = \frac{1}{\sqrt{5}} \quad (\text{since } A_1 + A_2 = 1\,)$$

Solve

$$\begin{array}{ll} A_1 + A_2 & = 1 \\ A_1 - A_2 & = \frac{\sqrt{5}}{5} \end{array} \quad\Rightarrow\quad \left\{\begin{array}{ll} A_1 = & \frac{5+\sqrt{5}}{10} \\ A_2 = & \frac{5-\sqrt{5}}{10} \end{array}\right.$$

So

$$T(n) = T_h(n) + T_p(n)$$
$$= \frac{5+\sqrt{5}}{10}\left(\frac{1+\sqrt{5}}{10}\right)^n + \frac{5-\sqrt{5}}{10}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

2. Solve

$$T(n) + 6T(n-1) + 12T(n-2) + 8T(n-3) = 0$$
$$T(n) = A\alpha^n$$
$$\Rightarrow \ \alpha^3 + 6\alpha^2 + 12\alpha + 8 = 0$$

is the characteristic equation.

$$\alpha^3 + 6\alpha^2 + 12\alpha + 8 = (\alpha+2)(\alpha^2 + 4\alpha + 4) = (\alpha+2)^3$$

So $\alpha = -2$ is a characteristic root of multiplicity 3.
And therefore

$$T_h(n) = (A_1 n^2 + A_2 n + A_3)(-2)^n$$

is a homogeneous solution. To determine A_1, A_2 and A_3 use the boundary conditions.
Remark: Recall $T(n) = T_h(n) + T_p(n)$. There is no general method for finding the particular solution. Inspection is a good technique that sometimes works.

In general, when the right-hand side of the recurrence relation is of the form n^t, the corresponding particular solution will be of the form:

$$P_1 n^t + P_2 n^{t-1} + \cdots + P_t n + P_{t+1}$$

where P_i's are constants.

3. Find the particular solution of:

$$T(n) + 5T(n-1) + 6T(n-2) = 3n^2$$

Solution:

$$\text{try } T_p(n) = P_1 n^2 + P_2 n + P_3$$
$$\Rightarrow \ P_1 n^2 + P_2 n + P_3 + 5P_1(n-1)^2 +$$
$$5P_2(n-1) + 5P_3 + 6P_1(n-2)^2 + 6P_2(n-2) + 6P_3 = 3n^2$$
$$\Rightarrow \ 12P_1 n^2 + (34P_1 - 12P_2)n + (29P_1 - 17P_2 + 12P_3) = 3n^2$$
$$\Rightarrow \ \begin{cases} 12P_1 & = 3 \\ 34P_1 - 12P_2 & = 0 \\ 29P_1 - 17P_2 + 12P_3 & = 0 \end{cases} \ \Rightarrow \ \begin{cases} P_1 & = \frac{1}{4} \\ P_2 & = \frac{17}{24} \\ P_3 & = \frac{115}{288} \end{cases}$$

Therefore

$$T_p(n) = \frac{1}{4}n^2 + \frac{17}{24}n + \frac{115}{288}\,.$$

When $f(n)$ is of the form of a polynomial of degree t:

$$F_1 n^t + F_2 n^{t-1} + \cdots + F_t n + F_{t+1}$$

then the corresponding solution will be of the form:

$$P_1 n^t + P_2 n^{t-1} + P_3 n^{t-2} + \cdots + P_t n + P_{t+1}\,.$$

In general:

Form of $f(n)$	Form of $T_p(n)$
β^n	$P\beta^n$, if β is **not** a characteristic root of the recurrence relation
$(F_1 n^t + F_2 n^{t-1} + \cdots + F_t n + F_{t+1})\beta^n$	$(P_1 n^t + P_2 n^{t-1} + \cdots + P_t n + P_{t+1})\beta^n$, if β is **not** a characteristic root of the recurrence relation AND $n^{m-1}(P_1 n^t + P_2 n^{t-1} + \cdots + P_t n + P_{t+1})\beta^n$ if β is a characteristic root of multiplicity $m-1$ of the recurrence relation.

EXAMPLES

1. Solve

$$T(n) + 5T(n-1) + 6T(n-2) = 42 \cdot 4^n$$
$$T(2) = 278$$
$$T(3) = 962$$

(a) **homogeneous solution**

$$T(n) = A\alpha^n \Rightarrow \ \alpha^3 + 5\alpha + 6 = 0$$

$$\Rightarrow \begin{cases} \alpha_1 = -2 \\ \alpha_2 = -3 \end{cases}$$

Therefore $T_h(n) = A_1(-2)^n + A_2(-3)^n$.

(b) **particular solution**

$$f(n) = 42 \cdot 4^n \text{ and 4 is not a chracteristic root.}$$

So try $T_p(n) = P4^n$.

$$P4^n + 5P4^{n-1} + 6P4^{n-2} = 42 \cdot 4^n$$

$$P4^{n-2}(4^2 + 5(4) + 6) = 42 \cdot 4^n$$

$$\frac{P4^n \cdot 42}{16} = 42 \cdot 4^n \ \Rightarrow \ P = 16$$

Therefore the particular solution is

$$T_p(n) = 16 \cdot 4^n$$

So the total solution is:

$$T(n) = T_h(n) + T_p(n) = A_1(-2)^n + A_2(-3)^n + 16 \cdot 4^n$$

(c) **use boundary conditions**

$$T(2) = 278\,, \ T(3) = 962$$

For $n = 2$:

$$T(2) + A_1 \cdot 4 + A_2 \cdot 9 + 16 \cdot 16$$

$$\text{So } \ 278 = 4A_1 + 9A_2 + 256 \qquad (1)$$

For $n = 3$:

$$T(3) = A_1(-2)^3 + A_2(-3)^3 + 16 \cdot 4^3$$

$$\Rightarrow 962 = -8A_1 - 27A_2 + 1024 \qquad (2)$$

Solving (1) and (2) we get:

$$A_1 = 1 \text{ and } A_2 = 2$$

So the solution is:

$$T(n) = (-2)^n + 2(-3)^n + 16 \cdot 4^n$$

2. Solve

$$\begin{array}{l} T(n) + T(n-1) = 3n2^n \\ T(0) = \frac{5}{3} \end{array}$$

   (a) **homogeneous solution**

$$T(n) = A\alpha^n \Rightarrow \begin{array}{l} A\alpha^n + A\alpha^{n-1} = 0 \\ A\alpha^{n-1}(\alpha + 1) \\ \alpha = -1 \end{array}$$

   (b) **particular solution**

$$f(n) = 3n2^n \textbf{ is of the form } (F_1 n + F_2)\beta^n$$

   where β is not a characteristic root. So try

$$T_p(n) = (P_1 n + P_2)2^n$$

$$\Rightarrow \ (P_1 n + P_2)2^n + [P_1(n-1) + P_2]2^{n-1} = 3n2^n$$

$$\Rightarrow \ 3P_1 n - P_1 + 3P_2 = 6n \ \Rightarrow \ 3P_1 = 6 \text{ and } 3P_2 - P_1 = 0$$

$$\Rightarrow P_1 = 2 \text{ and } P_2 = \frac{2}{3}$$

$$\Rightarrow \ T_p(n) = (2n + \frac{2}{3})2^n$$

   (c) **use boundary conditions**

$$\begin{aligned} T(n) \ &= T_h(n) + T_p(n) \\ &= A(-1)^n + (2n + \tfrac{2}{3})2^n \end{aligned}$$

$$T(0) = \frac{5}{3} \ \Rightarrow \ \frac{5}{3} = A + \frac{2}{3} \ \Rightarrow \ A = 1$$

   Therefore

$$T(n) = (-1)^n + (2n + \frac{2}{3})2^n$$

3. Solve

$$\begin{aligned} T(n) - 2T(n-1) &= 3 \cdot 2^2 \\ T(0) &= 1 \end{aligned}$$

   (a) **homogeneous solution:**

$$T(n) = A\alpha^n \Rightarrow \ A\alpha^n - 2A\alpha^{n-1} = 0$$

$$\Rightarrow \ A\alpha^{n-1}(\alpha - 2) = 0$$

$$\Rightarrow \ T_h(n) = A \cdot 2^n$$

(b) **particular solution**

$$f(n) = 3 \cdot 2^n \text{ and 2 is a characteristic root of multiplicity 1}$$

So try

$$T_p(n) = nP2^n$$

$$\Rightarrow \; Pn2^n - 2P(n-1)2^{n-1} = 3 \cdot 2^n \; \Rightarrow \; P = 3$$

So $T_p(n) = 3n(2)^n$.

(c) **use boundary value**

$$T(n) = T_h(n) + T_p(n) = A \cdot 2^n + 3n2^n$$

$$T(0) = 1 \; \Rightarrow \; 1 = A + 0 \; \Rightarrow \; A = 1$$

So $T(n) = 3n(2)^n + (2)^n = 2^n(3n+1)$.

4. Solve

$$\begin{aligned} T(n+2) &= 4T(n+1) - 4T(n) + 2^n \\ T(0) &= 2 \\ T(1) &= 4 \end{aligned}$$

(a) **homogeneous solution:**

$$T(n) = A\alpha^n \Rightarrow \; A\alpha^n(\alpha^2 - 4\alpha + 4) = 0 \Rightarrow (\alpha - 2)^2 = 0$$

$\alpha = 2$ is a root of multiplicity 2. So

$$T_h(n) = (A_1 n + A_2)2^n$$

(b) **particular solution:**

$$T(n+2) - 4T(n+1) + 4T(n) = 2^n$$

$f(n) = 2^n$ and 2 is a root of multiplicity 2. So try

$$T_p(n) = Pn^2 2^n$$

$$P(n+2)^2 2^{n+2} - 4P(n+1)^2 2^{n+1} + 4Pn^2 2^n = 2^n$$

$$\Rightarrow \; P = \frac{1}{8} \; \Rightarrow \; T_p(n) = n^2 2^{n-3}$$

(c) **use boundary conditions:**

$$T(n) = T_h(n) + T_p(n) = A_1 n 2^n + A_2 2^n + n^2 2^{n-3}$$

$$\begin{array}{ll} T(0) & = 2 \\ T(1) & = 4 \end{array} \Rightarrow \begin{array}{ll} A_1 & = 2 \\ A_2 & = -\frac{1}{8} \end{array}$$

Therefore $T(n) = 2^{n+1} - n2^{n-3} + n^2 2^{n-3}$.

5. Solve

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

$\alpha^2 - \alpha + 1 = 0$ is the characteristic equation.

$$\Rightarrow \ \alpha_1 = \frac{1-\sqrt{3}i}{2} \ \alpha_2 = \frac{1+\sqrt{3}i}{2}$$

$$T_h(n) = A_1\left(\frac{1-\sqrt{3}i}{2}\right)^n + A_2\left(\frac{1+\sqrt{3}i}{2}\right)^n$$

Note: $T(n) = T(n-1) - T(n-2) \Rightarrow T(2) = T(1) - T(0)$

$$\Rightarrow T(0) = 1$$

$$T_h(n) = A_1\left(\frac{1-\sqrt{3}i}{2}\right)^n + A_2\left(\frac{1+\sqrt{3}i}{2}\right)^n$$

$$T(1) = 1 \ \Rightarrow \ 1 = A_1\left(\frac{1-\sqrt{3}i}{2}\right) + A_2\left(\frac{1+\sqrt{3}i}{2}\right) \qquad (1)$$

$$T(0) = 1 \ \Rightarrow \ 1 = A_1 + A_2$$

$$\Rightarrow \ \frac{1+\sqrt{3}i}{2} = A_1\left(\frac{1+\sqrt{3}i}{2}\right) + A_2\left(\frac{1+\sqrt{3}i}{2}\right) \qquad (2)$$

$$(2)-(1) \ \Rightarrow \ \frac{1+\sqrt{3}i}{2} - 1 = A_1\left(\frac{1+\sqrt{3}i}{2} - \frac{1-\sqrt{3}i}{2}\right) + A_2 \cdot 0$$

$$\Rightarrow \ \frac{1+\sqrt{3}i-2}{2} = A_1\left(\frac{2\sqrt{3}i}{2}\right) \Rightarrow \ A_1 = \frac{(\sqrt{3}i-1)(\sqrt{3}i)}{(2\sqrt{3}i)(\sqrt{3}i)}$$

$$A_1 = \frac{3+\sqrt{3}i}{6} \ \Rightarrow A_2 = 1 - A_1 = \frac{3-\sqrt{3}i}{6}$$

So

$$T(n) = \left(\frac{3+\sqrt{3}i}{6}\right)\left(\frac{1-\sqrt{3}i}{2}\right)^n + \left(\frac{3-\sqrt{3}i}{6}\right)\left(\frac{1+\sqrt{3}i}{2}\right)^n$$

Note: We don't have a particular solution here.

6. Solve

$$R^2(n) - 2R^2(n-1) = 1$$

$$R(0) = 2$$

Note: $R^2(n)$ means $[R(n)]^2$.
Let $T(n) = R^2(n)$.

$$\begin{aligned} R^2(n) - 2R^2(n-1) &= 1 \\ R(0) &= 2 \end{aligned} \quad \Rightarrow \quad \begin{aligned} T(n) - 2T(n-1) &= 1 \\ T(0) = R^2(0) &= 4 \end{aligned}$$

$$T(n) = A\alpha^n \ \Rightarrow \ A\alpha^n - 2A\alpha^{n-1} = 0$$

$$A\alpha^{n-1}(\alpha - 2) = 0 \ \Rightarrow \ \alpha = 2$$

So $T_h(n) = A2^n$.

$$T_p(n) = B \qquad B - 2B = 1 \Rightarrow B = -1$$

So $T(n) = A2^n - 1$.

$$T(0) = 4 \Rightarrow 4 = A - 1 \Rightarrow A = 5$$

Therefore $T(n) = 5 \cdot 2^n - 1$. And $R(n) = \sqrt{5 \cdot 2^n - 1}$.

In the next section we show how generating functions can be used to solve recurrence relations.

2.5.2 Generating functions

To better understand generating functions, we start by defining **power series**.

$$\sum_{k=0}^{\infty} a_k x^k = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k + \cdots$$

is called a power series

- expansion about the origin
- think of a power series as a polynomial of infinite degree

If the expansion is about the point $x = a$, then the power series is a Taylor series and has the form

$$\sum_{k=0}^{\infty} a_k (x-a)^k = a_0 + a_1(x-a) + a_2(x-a)^2 + \cdots + a_k(x-a)^k + \cdots$$

Fact: The convergence set of a power series is always an interval with a as the midpoint of the interval of convergence. The interval of convergence might be reduced to a point, or may consist of all real numbers.

Fact: If the Taylor series $f(x) = \sum_{k=0}^{\infty} a_k(x-a)^k$ and all of the series obtained by differentiating it [i.e. $f'(x) = \sum_{k=1}^{\infty} k a_k x^{k-1}$, $f''(x) = \ldots$ etc.] converge for $a - \tau < x < a + \tau$ where $\tau > 0$, then:

$$a_k = \frac{f^{(k)}(a)}{k!} \text{ for } k = 0, 1, 2, \ldots$$

Examples:

① *Show that the geometric series* $\sum_{k=0}^{\infty} x^k$ *converges for* $-1 < x < 1$ *i.e. interval of convergence is the open interval:* $(-1; 1)$.

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} \qquad -1 < x < 1$$

$$\text{Let } f(x) = \frac{1}{1-x}, \; f'(x) = \frac{1}{(1-x)^2}$$

$$f''(x) = \frac{2}{(1-x)^3}, \; f^{(k)}(x) = \frac{k!}{(1-x)^{k+1}}$$

$$\text{So } \frac{f^{(k)}(0)}{k!} = \frac{k!}{k!(1-0)^{k+1}} = 1$$

$$\text{i.e. } a_k = 1 \text{ for } k = 0, 1, 2, \ldots$$

So $f(x) = f(0) + \frac{f'(0)}{1!}(x-0) + \frac{f''(0)}{2!}(x-0)^2 + \cdots + \frac{f^{(k)}(0)}{k!}(x-0)^k + \cdots$.

$$f(x) = 1 + x + x^2 + \cdots + x^k + \cdots$$

$$\text{i.e. } \frac{1}{1-x} = 1 + x + x^2 + \cdots + x^k + \cdots$$

□

② *Show that if* $f(x) = \frac{1}{(1-x)^2}$ *then* $a_k = k+1$. *In other words, show that*

$$\sum_{k=0}^{\infty}(k+1)x^k = \frac{1}{(1-x)^2}$$

$$f(x) = \frac{1}{(1-x)^2}\ ,\ f'(x) = \frac{2}{(1-x)^3}\ ,\ f''(x) = \frac{2\cdot 3}{(1-x)^4}$$

$$\text{and } f^{(k)}(x) = \frac{(k+1)!}{(1-x)^{k+2}}$$

So

$$a_k = \frac{f^{(k)}(0)}{k!} = \frac{(k+1)!}{k!(1-0)^{k+2}} = k+1$$

We showed that $a_k = k+1$, i.e.

$$\frac{1}{(1-x)^2} = \sum_{k=0}^{\infty}(k+1)x^k$$

Notation: $G(x) = \frac{1}{(1-x)^2}$.
$G(x)$ is called the **generating function** of a_0, a_1, a_2, … .
Here $G(x) = \frac{1}{(1-x)^2}$ is the generating function of 1, 2, 3, … , $k+1$, … .

□

Other method for tackling $G(x) = \frac{1}{(1-x)^2}$: We know

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

By differentiating both sides k times (and dividing by $k!$), we obtain:

$$\frac{1}{(1-x)^{k+1}} = 1 + (k+1)x + \frac{(k+1)(k+2)}{2!}x^2 + \cdots + \frac{(n+k)!}{k!n!}x^n + \cdots$$

Now let $k = 1$. □

③ *Show that if* $G(x) = (1+x)^r$ *then* $a_k = \binom{r}{k}$, *for any* $r \in \mathbf{R}$.

$$\begin{aligned}
G(x) &= (1+x)^r \\
G'(x) &= r(1+x)^{r-1} \\
G''(x) &= r(r-1)(1+x)^{r-2} \\
\vdots & \qquad \vdots \\
G^{(k)}(x) &= r(r-1)(r-2)\cdots(r-(k-1))(1+x)^{r-k} \\
&= \frac{[1\cdot 2\cdots(r-k-1)(r-k)](r-k+1)\cdots r}{1\cdot 2\cdot 3\cdots(r-k)}(1+x)^{r-k} \\
&= \frac{r!}{(r-k)!}(1+x)^{r-k}
\end{aligned}$$

$$\Rightarrow G^{(k)}(0) = \frac{r!}{(r-k)!}(1+0)^{r-k}$$

$$\text{And } \frac{G^{(k)}(0)}{k!} = \frac{r!}{k!(r-k)!}(1+0)^{r-k} \Rightarrow a_k = \binom{r}{k}$$

So $(1+x)^r = 1 + \binom{r}{1}x + \binom{r}{2}x^2 + \cdots + \binom{r}{k}x^k + \cdots$.
In other words: $G(x) = (1+x)^r$ is the generating function of:

$$\binom{r}{0},\ \binom{r}{1},\ \binom{r}{2},\ \ldots,\ \binom{r}{k},\ \ldots$$

□

④ *Show that* $G(x) = \ln\left(\frac{1}{1-x}\right)$ *is the generating function of :*

$$0\,,\ \frac{1}{2}\,,\ \frac{1}{3}\,,\ \ldots \frac{1}{k}\,,\ \ldots$$

$$\begin{aligned} G'(x) &= \frac{1}{1-x} \\ G''(x) &= \frac{1}{(1-x)^2} \\ G'''(x) &= \frac{2}{(1-x)^3} \\ &\vdots \\ G^{(k)}(x) &= \frac{(k-1)!}{(1-x)^k} \end{aligned}$$

$$\text{So } \frac{G^{(k)}(0)}{k!} = \frac{(k-1)!}{k!(1-0)^k}$$

i.e. $a_k = \frac{1}{k}$. And

$$\ln\left(\frac{1}{1-x}\right) = 0 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{k}x^k + \cdots$$

□

Other Method
We know:

$$1 + x + x^2 + x^3 + \cdots + x^n + \cdots = \frac{1}{1-x}$$

We integrate both sides to obtain the following:

$$x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \cdots + \frac{1}{n+1}x^{n+1} + \cdots = \int \frac{1}{1-x}dx$$

But $\int \frac{1}{1-x}dx = \ln \frac{1}{1-x}$.

$$\text{So } \begin{cases} a_0 &= 0 \\ a_k &= \frac{1}{k} \text{ for } k \geq 1 \end{cases}$$

□

⑤ *Show that the generating function of*

$$\left\{\begin{array}{ll} a_0 & = 0 \\ a_k & = \frac{1}{k} \text{ if } k \text{ is odd} \\ a_k & = -\frac{1}{k} \text{ if } k \text{ is even} \end{array}\right\} \quad \text{is } \ln(1+x)$$

We know

$$\begin{aligned} y + \frac{y^2}{2} + \frac{y^3}{3} + \cdots + \frac{y^n}{n} + \cdots &= \ln\left(\frac{1}{1-y}\right) \\ y + \frac{y^2}{2} + \frac{y^3}{3} + \cdots + \frac{y^n}{n} + \cdots &= -\ln(1-y) \\ -y - \frac{y^2}{2} - \frac{y^3}{3} - \cdots - \frac{y^n}{n} - \cdots &= \ln(1-y) \end{aligned}$$

Let $y = -x$.

$$\Rightarrow \; x - \frac{x^2}{2} \pm \cdots = \ln(1+x)$$

So

$$\begin{gathered} a_0 = 0 \\ a_k = \tfrac{1}{k} \text{ if } k \text{ is odd} \\ a_k = -\tfrac{1}{k} \text{ if } k \text{ is even} \end{gathered}$$

has $G(x) = \ln(1+x)$ as generating function.

□

Other method:

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

$$\Rightarrow \; \frac{1}{1-(-x)} = 1 + (-x) + (-x)^2 \mp \cdots + (-x)^k + \cdots$$

$$\Rightarrow \; \frac{1}{1-(-x)} = \sum_{k=0}^{\infty} (-x)^k$$

Integrating both sides, we get:

$$\begin{aligned} \ln(1+x) &= \int_0^x \frac{dx}{1+x} \\ &= \int_0^x \sum_{k=0}^{\infty} (-x)^k dx \\ = \left(\begin{array}{c}\text{since convergence is uniform} \\ \text{inside closed intervals}\end{array}\right) &= \sum_{k=0}^{\infty} \int_0^x (-x)^k dx \\ &= \sum_{k=0}^{\infty} \frac{(-1)^k x^{k+1}}{k+1} \\ &= \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^k}{k} \\ &= -\sum_{k=1}^{\infty} \frac{(-x)^k}{k} \end{aligned}$$

i.e.

$$\ln(1+x) = 0 + x - \frac{x^2}{2} \pm \cdots + \frac{(-x)^k}{k} \cdots$$

which is what we wanted to show. □

⑥ *$G(x) = e^x$, $G^k(x) = e^x$ for all k.*

$$\Rightarrow \; G^k(0) = e^0 = 1$$

Therefore $a_k = \frac{G^{(k)}(0)}{k!} = \frac{1}{k!}$.
And e^x is the generating function of 1, 1, $\frac{1}{2!}$, $\frac{1}{3!}$, ... In other words:

$$e^x = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots + \frac{1}{k!}x^k + \cdots$$

□

⑦ *Show that e^{rx} is the generating function of*

$$1\,,\;\; r\,,\;\; \frac{r^2}{2!}\,,\;\; \frac{r^3}{3!}\,,\;\; \cdots\,,\;\; \frac{r^k}{k!}\,,\;\; \cdots$$

We have:

$$1 + y + \frac{y^2}{2!} + \frac{y^3}{3!} + \cdots + \frac{y^n}{n!} + \cdots = e^y$$

Let $y = rx$.

$$\Rightarrow \; 1 + rx + \frac{(rx)^2}{2!} + \frac{(rx)^3}{3!} + \cdots + \frac{(rx)^n}{n!} + \cdots = e^{rx}$$

$$\Rightarrow \; 1 + rx + \frac{r^2}{2!}x^2 + \frac{r^3}{3!}x^3 + \cdots + \frac{r^n}{n!}x^n + \cdots = e^{rx}$$

So the generating function of

$$1\,,\;\; r\,,\;\; \frac{r^2}{2!}\,,\;\; \frac{r^3}{3!}\,,\;\; \ldots\,,\;\; \frac{r^k}{k!}\,,\;\; \ldots$$

is e^{rx}. □

The seven examples are summarized in the following table:

Some common generating functions

	Sequence a_k, $k = 0, 1, 2, \ldots$	Generating function $G(x)$
①	$a_k = 1$	$\frac{1}{1-x}$
②	$a_k = k + 1$	$\frac{1}{(1-x)^2}$
③	$a_k = \binom{r}{k}$, $r \in \mathbf{R}$	$(1+x)^r$
④	$a_0 = 0$, $a_k = \frac{1}{k}$ if $k \geq 1$	$\ln \frac{1}{1-x}$
⑤	$a_0 = 0$ $a_k = \frac{1}{k}$ if k is odd $a_k = -\frac{1}{k}$ if k is even	$\ln(1+x)$
⑥	$a_k = \frac{1}{k!}$	e^x
⑦	$a_k = \frac{r^k}{k!}$, $r \in \mathbf{R}$	e^{rx}

Operations on Generating Functions

$F(x)$ is the generating function of $a_0, a_1, a_2, \ldots, a_k, \ldots$
and $G(x)$ is the generating function of $b_0, b_1, \ldots, b_k, \ldots$.
In other words:

$$F(x) = \sum a_k x^k \text{ and } G(x) = \sum b_k x^k$$

operation on generating function	k-th element of the corresponding sequence
Multiplication by constant: cF	ca_k
Addition: $F+G$	$a_k + b_k$
Convolution: $F \cdot G$	$\sum_{i=0}^{k} a_i b_{k-i}$
Right shift: xF	a_{k-1}
Partial Sum: $F(x)/(1-x)$	$\sum_{i=0}^{k} a_i$
Differentiation (Index multiply): $x\frac{dF}{dx}(x)$	ka_k
Integration (Index divide): $\int_0^x F(t)dt$	a_{k-1}/k
Scaling: $F(cx)$	$c^k a_k$

Show that the sequence

$$t_k = \sum_{i=0}^{k} a_i b_{k-i}\ , \quad k = 0, 1, 2, \ldots$$

has generating function $F(x)G(x)$. This operation is called convolution (see table).
Solution:

$$\begin{aligned}
& t_0 + t_1 x + t_2 x^2 + \cdots + t_k x^k + \cdots \\
= \quad & a_0 b_0 + (a_0 b_1 + b_0 a_1)x + (a_0 b_2 + a_1 b_1 + a_2 b_0)x^2 \\
& + \cdots + (a_0 b_k + a_1 b_{k-1} + \cdots + a_i b_{k-i} + \cdots + a_k b_0)x^k + \cdots \\
= \quad & a_0(b_0 + b_1 x + b_2 x^2 + \cdots + b_k x^k + \cdots) \\
& + a_1 x(b_0 + b_1 x + b_2 x^2 + \cdots + b_k x^k + \cdots) \\
& \vdots \\
& + a_k x^k (b_0 + b_1 x + b_2 x^2 + \cdots + b_k x^k + \cdots) \\
& \vdots \\
= \quad & (a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k + \cdots)(b_0 + b_1 x + b_2 x^2 + \cdots + b_k x^k \cdots) \\
= \quad & F(x)G(x)
\end{aligned}$$

□

2.5.3 Using generating functions to solve recurrence relations

① Solve

$$a_n = 4a_{n-1}$$
$$a_0 = 1$$

$$\begin{aligned} G(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + \cdots \\ 4G(x) &= 4a_0 + 4a_1x + 4a_2x^2 + \cdots + 4a_nx^n + \cdots \end{aligned}$$

$$\begin{aligned} \Rightarrow\ G(x) - 4xG(x) &= a_0 + (a_1 - 4a_0)x + (a_2 - 4a_1)x^2 + \\ &\quad + \cdots + (a_n - 4a_{n-1})x^n + \cdots \end{aligned}$$

But $a_0 = 1$ and $a_n - 4a_{n-1} = 0 \quad \forall n.$

$$\Rightarrow\ G(x)[1-4x] = 1$$

$$\Rightarrow\ G(x) = \frac{1}{1-4x}$$

We know that

$$1 + x + x^2 + x^3 + \cdots + x^n + \cdots = \frac{1}{1-x}$$

$$\Rightarrow\ \frac{1}{1-4x} = \sum_{n=0}^{\infty}(4x)^n = \sum_{n=0}^{\infty} 4^n x^n$$

$$\Rightarrow\ a_n = 4^n \text{ which is the coefficient of the } n^{th} \text{ term.}$$

Solve

$$a_n - 3a_{n-1} + 2a_{n-2} = 2^n$$
$$a_0 = 0\ ,\ \ a_1 = 1$$

$$\begin{aligned} G(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + \cdots \\ 3xG(x) &= 3a_0x + 3a_1x^2 + 3a_2x^3 + \cdots + 3a_{n-1}x^n + \cdots \\ 2x^2G(x) &= 2a_0x^2 + 2a_1x^3 + 2a_2x^4 + \cdots + 2a_{n-2}x^n + \cdots \end{aligned}$$

$$\Rightarrow\ G(x) - 3xG(x) + 2x^2G(x)$$

$$= a_0 + (a_1 - 3a_0)x + (a_2 - 3a_1 + 2a_0)x^2 + \ \cdots + (a_n - 3a_{n-1} + 2a_{n-2})x^n + \cdots$$

But $a_0 = 0$, $a_1 - 3a_0 = 1$, $a_2 - 3a_1 + 2a_0 = 2^2$ and $a_n - 3a_{n-1} + 2a_{n-2} = 2^n$.

$$\Rightarrow\ (1 - 3x + 2x^2)G(x) = x + 2^2x^2 + 2^3x^3 + \cdots + 2^nx^n + \cdots$$

$$= x + \sum_{n=2}^{\infty} 2^n x^n$$

$$= x + \left[\sum_{n=0}^{\infty}(2x)^n - (2x)^0 - (2x)^1\right]$$

$$= x + \left[\frac{1}{1-2x} - 1 - 2x\right]$$

$$\begin{aligned}
(1-3x+2x^2)G(x) &= x + \left(\frac{1-1+2x-2x+4x^2}{1-2x}\right) \\
&= x + \frac{4x^2}{1-2x} \\
(1-x)(1-2x)G(x) &= x + \frac{4x^2}{1-2x}
\end{aligned}$$

$$\Rightarrow \; G(x) = \frac{x}{(1-x)(1-2x)} + \frac{4x^2}{(1-x)(1-2x)^2}$$

Solve

$$\frac{x}{(1-x)(1-2x)} = \frac{A}{1-x} + \frac{B}{1-2x} \text{ to get } A = -1\,, \;\; B = 1$$

and solve

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

$$\Rightarrow \; A = 1\,, \;\; B = -\frac{3}{2}\,, \;\; C = \frac{1}{2}$$

Therefore

$$G(x) = \left(\frac{1}{1-2x} - \frac{1}{1-x}\right) + 4\left[\frac{1}{1-x} - \frac{3}{2(1-2x)} + \frac{\frac{1}{2}}{(1-2x)^2}\right]$$

$$\begin{aligned}
\frac{1}{1-x} &= \sum_{i=0}^{\infty} x^i \;\; \Rightarrow \;\; a_n = 1 \\
\frac{1}{1-2x} &= \sum_{i=0}^{\infty} (2x)^i \;\; \Rightarrow \;\; a_n = 2^n \\
& \quad \text{differentiating both sides} \\
\frac{2}{(1-2x)^2} &= \sum_{i=0}^{\infty} (i+1)2^{i+1}x^i \\
\Rightarrow \;\; \frac{1}{(1-2x)^2} &= \sum_{i=0}^{\infty} (i+1)2^i x^i
\end{aligned}$$

So

$$G(x) = \frac{1}{1-2x} - \frac{1}{1-x} + 4\left(\frac{1}{1-x} - \frac{\frac{3}{2}}{1-2x} + \frac{\frac{1}{2}}{(1-2x)^2}\right)$$

is the generating function of a sequence whose n^{th} term is:

$$\begin{aligned}
a_n &= 2^n - 1 + 4\left(1 - \frac{3}{2}(2^n) + \frac{1}{2}(n+1)2^n\right) \\
&= (1 - 6 + 2n + 2)2^n - 1 + 4
\end{aligned}$$

So

$$a_n = (2n-3)2^n + 3$$

is the solution of

$$a_n - 3a_{n-1} + 2a_{n-2} = 2^n \text{ with } a_0 = 0\,,\ \ a_1 = 1$$

□

② Use generating functions to solve

$$\begin{aligned} a_n &= 4a_{n-2} \\ a_0 &= 0 \\ a_1 &= 1 \end{aligned}$$

$$\begin{aligned} G(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_nx^n + \cdots \\ 4x^2G(x) &= 4a_0x^2 + 4a_1x^3 + 4a_2x^4 + \\ &\quad \cdots + 4a_nx^{n+2} + \cdots \\ (1-4x^2)G(x) &= a_0 + a_1x + (a_2 - 4a_0)x^2 + \\ &\quad \cdots + (a_n - 4a_{n-2})x^n + \cdots \\ (1-4x^2)G(x) &= a_0 + a_1x + \sum_{i=2}^{\infty}(a_i - 4a_{i-2})x^i \end{aligned}$$

But $a_0 = 0$, $a_1 = 1$ and $a_i - 4a_{i-2} = 0$.

$$G(x) = \frac{x}{1-4x^2} = \frac{x}{(1-2x)(1+2x)}$$

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

$$\Rightarrow\ A = \frac{1}{4}\,,\ \ B = -\frac{1}{4}$$

$$G(x) = \frac{\frac{1}{4}}{1-2x} + \frac{-\frac{1}{4}}{1+2x}$$

We know

$$\frac{1}{1-y} = 1 + y + y^2 + y^3 + \cdots + y^n + \cdots$$

Hence with $y = 2x$ we get:

$$\frac{1}{1-2x} = 1 + 2x + 2^2x^2 + 2^3x^3 + \cdots + 2^nx^n + \cdots$$

and with $y = -2x$ we obtain:

$$\frac{1}{1-(-2x)} = 1 + (-2)x + (-2)^2x^2 + (-2)^3x^3 + \cdots + (-2)^nx^n + \cdots$$

$$\Rightarrow\ a_n = \frac{1}{4}2^n - \frac{1}{4}(-1)^n2^n$$

Therefore

$$\text{for } n \text{ even} \ \ \Rightarrow\ a_n = 0$$

$$\text{for } n \text{ odd} \ \ \Rightarrow\ a_n = 2^{n-1}$$

□

③ Find the generating function of the sequence of cubes:

$$0, 1, 8, 27, 64, \ldots, n^3, \ldots$$

Solution:

$$1 + x + x^2 + x^3 + \ldots + x^n + \ldots = \frac{1}{1-x}$$

differentiate both sides:

$$1 + 2x + 3x^2 + 4x^3 + \ldots + nx^{n-1} + \ldots = \frac{1}{(1-x)^2}$$

multiply by x:

$$x + 2x^2 + 3x^3 + 4x^4 + \ldots nx^n + \ldots = \frac{x}{(1-x)^2}$$

differentiate both sides:

$$1 + 2^2x + 3^2x^2 + 4^2x^3 + \ldots n^2x^{n-1} + \ldots = \frac{1+x}{(1-x)^3}$$

multiply by x:

$$x + 2^2x^2 + 3^2x^3 + 4^2x^4 + \ldots n^2x^n + \ldots = \frac{x+x^2}{(1-x)^3}$$

differentiate both sides:

$$1 + 2^3x + 3^3x^2 + 4^3x^3 + \ldots n^3x^{n-1} + \ldots = \frac{x^2+4x+1}{(1-x)^4}$$

multiply by x:

$$0 + x + 2^3x^2 + 3^3x^3 + 4^3x^4 + \ldots n^3x^n + \ldots = \frac{x(x^2+4x+1)}{(1-x)^4}$$

So

$$G(x) = \frac{x(x^2+4x+1)}{(1-x)^4}$$

is the generating function of $0, 1, 8, 27, 64, \ldots, n^3, \ldots$.

$\square$

Bibliography

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* MIT Press, Cambridge, 1990.

[Liu68] C.L. Liu. *Introduction to Combinatorial Mathematics.* McGraw-Hill, 1968.

[Liu85] C.L. Liu. *Elements of Discrete Mathematics.* McGraw-Hill, 1985.