CSE 317: Design and Analysis of Algorithms

Shahid Hussain

Recurrences: Fall 2025

Sequences and Series

Sequences

Sequence

A sequence is an enumerated collection of objects. For example:

n	0	1	2	3	4	5	6	...
a_n	0	1	1	2	3	5	8	...

is the famous sequence of *Fibonacci* numbers.

Sequences

Arithmetic Sequence

A sequence is called an arithmetic progression (or sequence) if it is of the form:

$$a, a+d, a+2d, \ldots, a+nd, \ldots$$

where a is the initial term and d is the common difference, $a, d \in \mathbb{R}$.

Geometric Sequence

A sequence is called a geometric progression (or sequence) if it is of the form:

$$a, ar, ar^2, \ldots, ar^n, \ldots$$

where a is the initial term and r is the common ration, $a, r \in \mathbb{R}$.

Examples

Arithmetic sequences:

- $[a = 1, d = 1]$ $1, 2, 3, 4, \ldots$
- $[a = 0, d = 2]$ $0, 2, 4, 6, \ldots$
- $[a = 10, d = -6]$ $10, 4, -2, -8, \ldots$

Geometric sequences:

- $[a = 1, r = -1]$ $1, -1, 1, -1, \ldots$
- $[a = 1, r = 1/2]$ $1, 1/2, 1/4, 1/8, \ldots$
- $[a = 1, r = 2]$ $2^0, 2^1, 2^2, 2^3, \ldots$

Recurrences

Recurrences

- Recurrences (or recurrence relations or difference equations) arise naturally during analysis of algorithms
- For example, most recursive algorithms can be represented by a recurrence and then the time complexity of the algorithm is just the solution of the recurrence
- For example consider the following recursive algorithm that computes n! for any $n \geq 1$

Algorithm: FACTORIAL

Input: An integer $n \geq 1$

Output: The value of $n!$

1. **if** $n = 1$ **then return** 1
2. **else return** $n * \text{FACTORIAL}(n - 1)$

Recurrences

- Assume that the time to compute factorial of an integer $n \geq 1$ is represented as $T(n)$
- Clearly $T(n) = 1 + T(n-1)$, furthermore, $T(1) = 1$
- We can solve this recurrence as following:

$$\begin{aligned} T(n) &= 1 + T(n-1) \\ &= 1 + 1 + T(n-2) \\ &= 1 + 1 + 1 + T(n-3) \\ & \quad \ldots \\ &= 1 + 1 + 1 + \cdots + 1 + T(1) \\ &= n = \Theta(n) \end{aligned}$$

A Simple Recurrence

- Consider the following recurrence:

$$A(n) = 2A(n-1) + 1, \qquad \text{such that} \quad A(0) = 0$$

- First few terms of the sequence are: $0, 1, 3, 7, 15, 31, 63, \ldots$
- It seems like $A(n) = 2^n - 1$
- We can prove this by induction
- Base case: $A(0) = 0 = 2^0 - 1$
- Inductive step: Assume that $A(k) = 2^k - 1$ for all $k < n$
- Then for $n = k + 1$ we have:

$$A(k+1) = 2A(k) + 1 = 2(2^k - 1) + 1 = 2^{k+1} - 2 + 1 = 2^{k+1} - 1$$

A Simple Recurrence

- We can also solve this recurrence by expanding the recursive term as follows:

$$\begin{aligned}
A(n) &= 2A(n-1) + 1 \\
&= 2(2A(n-2) + 1) + 1 \\
&= 2^2 A(n-2) + 2 + 1 \\
&= 2^2(2A(n-3) + 1) + 2 + 1 \\
&= 2^3 A(n-3) + 2^2 + 2 + 1 \\
&\quad \ldots \\
&= 2^n A(0) + 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2 + 1 \\
&= 2^n - 1 = \Theta(2^n)
\end{aligned}$$

A General Two Terms Recurrence

One Term Recurrence with Constant Coefficient

A recurrence of the following form, where a and $c > 0$ are constants and $f(n)$ is a function of n:

$$A(n) = cA(n-1) + f(n), \qquad \text{such that} \quad A(0) = a$$

has the solution:

$$A(n) = a \cdot c^n + \sum_{j=0}^{n-1} c^j \cdot f(n-j).$$

Proof for Two Terms Recurrences

$$\begin{aligned}
A(n) &= cA(n-1) + f(n) \\
&= c(cA(n-2) + f(n-1)) + f(n) \\
&= c^2 A(n-2) + cf(n-1) + f(n) \\
&= c^2(cA(n-3) + f(n-2)) + cf(n-1) + f(n) \\
&= c^3 A(n-3) + c^2 f(n-2) + cf(n-1) + f(n) \\
&\quad \ldots \\
&= c^n A(0) + c^{n-1} f(1) + c^{n-2} f(2) + \cdots + cf(n-1) + f(n) \\
&= a \cdot c^n + \sum_{j=0}^{n-1} c^j \cdot f(n-j).
\end{aligned}$$

Linear Homogeneous Recurrences

- A *linear homogeneous recurrence of degree k* with constant Coefficients is a recurrence relation which has k recursive terms and is of the form:

$$A(n) = c_1 A(n-1) + c_2 A(n-2) + \cdots + c_k A(n-k)$$

where c_i's are constants and $c_k \neq 0$ with $A(0) = C_0, \ldots, A(k-1) = C_{k-1}$ as k *initial conditions*
- We can solve this recurrence by finding the roots of the corresponding characteristic equation

Linear Homogeneous Recurrences

- We can rewrite the recurrence as follows:

$$A(n) - c_1 A(n-1) - c_2 A(n-2) - \cdots - c_k A(n-k) = 0$$

- The corresponding characteristic equation is:

$$x^k - c_1 x^{k-1} - c_2 x^{k-2} - \cdots - c_k = 0$$

- The roots of this equation $r_1, r_2, \ldots, r_k$ will be used to form the general solution of the recurrence
- The general solution of the recurrence is:

$$A(n) = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots + \alpha_k r_k^n$$

The Fibonacci Recurrence

- Let $f_n = f_{n-1} + f_{n-2}$ with $f_0 = 0$ and $f_1 = 1$ $(n \geq 0)$
- We can rewrite it as: $f_n - f_{n-1} - f_{n-2} = 0$
- The characteristic equation is: $x^2 - x - 1 = 0$
- The two real roots of this equation are: $\frac{1 \pm \sqrt{5}}{2}$
- Thus the general solution of the Fibonacci recurrence is:

$$f_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$$

- We can find the exact values of α_1 and α_2 by using the initial conditions, $\alpha_1 = \frac{1}{\sqrt{5}}$ and $\alpha_2 = -\frac{1}{\sqrt{5}}$

Another Three Term Recurrence

- Consider the recurrence: $a_n = a_{n-1} - a_{n-2}$ with $a_0 = 0$ and $a_1 = 1$ $(n \geq 2)$
- The first few terms of the sequence are: $0, 1, 1, 0, -1, -1, 0, \ldots$
- The characteristic equation is: $x^2 - x + 1 = 0$ which has the roots x_1, x_2:

$$x_1 = \frac{1}{2} + i\frac{\sqrt{3}}{2}, \quad x_2 = \frac{1}{2} - i\frac{\sqrt{3}}{2}$$

- We can rewrite these complex numbers in polar coordinates as $re^{i\theta}$
- In this case

$$r = \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} = 1 \quad \text{and} \quad \theta = \arctan\left(\frac{\sqrt{3}}{2} \times 2\right) = \frac{\pi}{3}$$

- $x_1 = e^{i\pi/3}$ and $x_2 = e^{-i\pi/3}$

Another Three Term Recurrence

- Let us find x_1^n and x_2^n:

$$x_1^n = \left(e^{i\pi/3}\right)^n = \cos\left(\frac{n\pi}{3}\right) + i\sin\left(\frac{n\pi}{3}\right)$$
$$x_2^n = \left(e^{-i\pi/3}\right)^n = \cos\left(\frac{n\pi}{3}\right) - i\sin\left(\frac{n\pi}{3}\right)$$

- We can thus find the general solution as

$$\begin{aligned} a_n &= \alpha_1 x_1^n + \alpha_2 x_2^n \\ &= \alpha_1\left[\cos\left(\frac{n\pi}{3}\right) + i\sin\left(\frac{n\pi}{3}\right)\right] + \alpha_2\left[\cos\left(\frac{n\pi}{3}\right) - i\sin\left(\frac{n\pi}{3}\right)\right] \end{aligned}$$

- We will use the initial conditions $a_0 = 0$ and $a_1 = 1$ to find α_1 and α_2

Another Three Term Recurrence

- We know that $a_0 = 0$ therefore:

$$\begin{aligned} 0 &= \alpha_1(\cos 0 + i\sin 0) + \alpha_2(\cos 0 - i\sin 0) \\ 0 &= \alpha_1 + \alpha_2 \quad \Longrightarrow \quad \alpha_1 = -\alpha_2 \end{aligned}$$

- Similarly, we know that $a_1 = 1$ therefore:

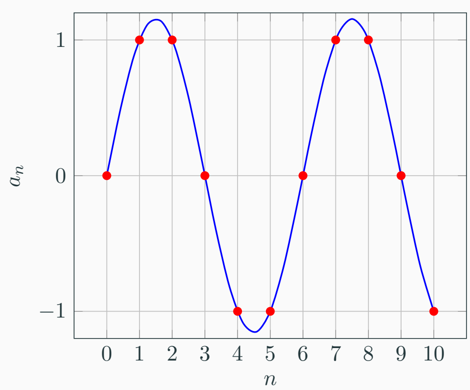
$$\begin{aligned} 1 &= \alpha_1\left[\cos\left(\frac{\pi}{3}\right) + i\sin\left(\frac{\pi}{3}\right)\right] + \alpha_2\left[\cos\left(\frac{\pi}{3}\right) - i\sin\left(\frac{\pi}{3}\right)\right] \\ &= \alpha_1\left(\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) + \alpha_2\left(\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \\ &= (\alpha_1 + \alpha_2)\frac{1}{2} + i\frac{\sqrt{3}}{2}(\alpha_1 - \alpha_2) \\ &= (\alpha_1 - \alpha_1)\frac{1}{2} + i\frac{\sqrt{3}}{2}(\alpha_1 + \alpha_1) = i\sqrt{3}\alpha_1 \quad \Longrightarrow \alpha_1 = -\frac{i}{\sqrt{3}} \end{aligned}$$

Another Three Term Recurrence

- Therefore the general solution of the recurrence is:

$$\begin{aligned} a_n &= \frac{-i}{\sqrt{3}}\left[\cos\left(\frac{\pi n}{3}\right) + i\sin\left(\frac{\pi n}{3}\right)\right] + \frac{i}{\sqrt{3}}\left[\cos\left(\frac{\pi n}{3}\right) - i\sin\left(\frac{\pi n}{3}\right)\right] \\ a_n &= \frac{2}{\sqrt{3}}\sin\left(\frac{\pi n}{3}\right) \end{aligned}$$

Discrete plot for the recurrence



The Problem: Linear Recurrence Relations

We start with a linear homogeneous recurrence relation of order k:

Recurrence Relation

$A(n) = c_1A(n-1) + c_2A(n-2) + \cdots + c_kA(n-k)$

The key is to solve its corresponding characteristic equation:

Characteristic Equation

$x^k - c_1x^{k-1} - c_2x^{k-2} - \cdots - c_k = 0$

The form of the general solution for $A(n)$ depends entirely on the nature of this equation's roots.

Types of Solutions

The solution's structure is determined by the types of roots found. There are three main cases:

1. Distinct Real Roots
2. Repeated Real Roots
3. Complex Conjugate Roots

Case 1: Distinct Real Roots

Condition: The characteristic equation has k distinct, real roots: $x_1, x_2, \ldots, x_k$.

General Solution

The solution is a simple linear combination of each root raised to the power of n:

$$A(n) = \alpha_1 x_1^n + \alpha_2 x_2^n + \cdots + \alpha_k x_k^n$$

The constants α_i are found using the initial conditions.

Example: Fibonacci Sequence

For $F(n) = F(n-1) + F(n-2)$, the equation is $x^2 - x - 1 = 0$. The roots are distinct and real: $x_{1,2} = \frac{1\pm\sqrt{5}}{2}$. The solution is $F(n) = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$.

Case 2: Repeated Real Roots

Condition: A real root, x_1, is repeated with a multiplicity of m.

General Solution Term

The contribution of this root to the solution is x_1^n multiplied by a polynomial in n of degree $m-1$:

$$(\alpha_1 + \alpha_2 n + \alpha_3 n^2 + \cdots + \alpha_m n^{m-1})x_1^n$$

Example

For $A(n) = 6A(n-1) - 9A(n-2)$, the equation is $x^2 - 6x + 9 = 0$, or $(x-3)^2 = 0$. The root is $x_1 = 3$ with multiplicity 2. The solution is $A(n) = (\alpha_1 + \alpha_2 n)3^n$.

Case 3: Complex Conjugate Roots

Condition: The roots include a complex conjugate pair, $a \pm bi$. We express this in polar form: $r(\cos\theta \pm i \sin\theta)$.

General Solution Term

This pair contributes an oscillating term to the solution, which can be written using real functions:

$$r^n(\alpha_1 \cos(n\theta) + \alpha_2 \sin(n\theta))$$

(If the pair has multiplicity m, the constants α_1, α_2 become polynomials in n of degree $m-1$).

Example

For $A(n) = 2A(n-1) - 2A(n-2)$, the equation is $x^2 - 2x + 2 = 0$. The roots are $1 \pm i$. In polar form, $r = \sqrt{2}$ and $\theta = \pi/4$. The solution is $A(n) = (\sqrt{2})^n \left(\alpha_1 \cos\left(\frac{n\pi}{4}\right) + \alpha_2 \sin\left(\frac{n\pi}{4}\right)\right)$.

Summary of Solutions

The general solution is a sum of terms, one for each root of the characteristic equation.

- **Distinct Real Root (x_1):**
  Contributes $\to \alpha_1 x_1^n$
- **Repeated Real Root (x_1, multiplicity m):**
  Contributes $\to (\alpha_1 + \cdots + \alpha_m n^{m-1})x_1^n$
- **Complex Pair ($r(\cos\theta \pm i\sin\theta)$):**
  Contributes $\to r^n(\beta_1 \cos(n\theta) + \beta_2 \sin(n\theta))$

Final Step

In all cases, the unknown constants (α_i, β_i) are found by solving a system of linear equations derived from the initial conditions of the recurrence (e.g., $A(0), A(1), \ldots$).

Divide-and-Conquer Recurrences

The General Form

Many divide-and-conquer algorithms have a running time expressed by the recurrence:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

where $T(1) = c$ for some constant c.

- $a \geq 1$ is the number of subproblems.
- $b > 1$ is the factor by which the input size is reduced.
- $f(n)$ is the cost of dividing the problem and combining the results of the subproblems.

Solution

The Master Theorem provides a "cookbook" method for finding the asymptotic bound of such recurrences.

The Core Idea

The theorem works by comparing the function $f(n)$ with a critical function, $n^{\log_b a}$.

Compare $f(n)$ vs. $n^{\log_b a}$

This comparison tells us which part of the algorithm dominates the runtime:

- The work done creating subproblems at each level ($f(n)$).
- The work done in the subproblems themselves ($aT(n/b)$), which culminates at the leaves of the recursion tree.

The Master Theorem: The Three Cases

Let $T(n) = aT\left(\frac{n}{b}\right) + f(n)$. We compare $f(n)$ with $n^{\log_b a}$.

Case 1: Leaf-Heavy

If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$. Then $\boldsymbol{T(n) = \Theta(n^{\log_b a})}$

Case 2: Balanced

If $f(n) = \Theta(n^{\log_b a})$. Then $\boldsymbol{T(n) = \Theta(n^{\log_b a} \log n)}$

Case 3: Root-Heavy

If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some $\epsilon > 0$, and the *regularity condition* $af(n/b) \leq kf(n)$ holds for some constant $k < 1$. Then $\boldsymbol{T(n) = \Theta(f(n))}$

Example: Merge Sort

Let's analyze the recurrence for Merge Sort.

Recurrence

$T(n) = 2T(n/2) + \Theta(n)$

1. **Identify parameters:**
   $a = 2$, $b = 2$, $f(n) = \Theta(n)$.
2. **Calculate the critical function:**
   $n^{\log_b a} = n^{\log_2 2} = n^1 = n$.
3. **Compare and determine the case:**
   We compare $f(n) = \Theta(n)$ with n. Since $f(n) = \Theta(n^{\log_b a})$, we are in **Case 2**.
4. **State the solution:**
   According to Case 2, the final complexity is:

$$T(n) = \Theta(n^{\log_2 2} \log n) = \Theta(n \log n)$$

Summary

Table 1: Master Theorem Quick Reference

Case	**Condition**	**Result**
Case 1	$f(n) = O(n^{\log_b a - \epsilon})$	$T(n) = \Theta(n^{\log_b a})$
Case 2	$f(n) = \Theta(n^{\log_b a})$	$T(n) = \Theta(n^{\log_b a} \log n)$
Case 3	$f(n) = \Omega(n^{\log_b a + \epsilon})$ + regularity	$T(n) = \Theta(f(n))$

The Master Theorem is a powerful and efficient tool for analyzing a wide range of divide-and-conquer algorithms.

The Akra-Bazzi Method?

A More Powerful Tool

The Akra-Bazzi method is a powerful generalization of the Master Theorem for analyzing the asymptotic behavior of divide-and-conquer recurrences.

Key Advantage

Its primary advantage is its ability to solve recurrences where the subproblems are not of uniform size.

General Form

The method applies to recurrences of the form:

$$T(n) = \sum_{i=1}^{k} a_i T\left(\frac{n}{b_i}\right) + f(n)$$

- k is the number of recursive terms.
- $a_i > 0$ are constants representing the number of subproblems of each type.
- $b_i > 1$ are constants representing the divisor for each subproblem size.
- $f(n)$ is the non-recursive cost function.

The Method: Step 1

Find the Unique Value 'p'

First, you must find the unique real number p that satisfies the characteristic equation:

$$\sum_{i=1}^{k} \frac{a_i}{b_i^p} = 1$$

This value, p, captures the dominant exponent of the recurrence.

The Method: Step 2

Calculate the Final Solution

Once p is found, the solution to the recurrence is given by the formula:

$$T(n) = \Theta\left(n^p\left(1 + \int_1^n \frac{f(x)}{x^{p+1}} dx\right)\right)$$

This integrates the contribution of the non-recursive work $f(n)$ over the problem sizes.

Example Analysis

Let's solve a recurrence the Master Theorem can't handle:

$$T(n) = 2T(n/4) + 3T(n/6) + \Theta(n \log n)$$

1. Find p

Solve the equation: $\frac{2}{4^p} + \frac{3}{6^p} = 1$.

By inspection, we find that $\boldsymbol{p = 1}$ is the solution: $\frac{2}{4} + \frac{3}{6} = \frac{1}{2} + \frac{1}{2} = 1$.

2. Calculate the Integral

With $p = 1$ and $f(x) = x \log x$, we evaluate:

$$\int_1^n \frac{x \log x}{x^{1+1}} dx = \int_1^n \frac{\log x}{x} dx = \left[\frac{1}{2}(\log x)^2\right]_1^n = \frac{1}{2}(\log n)^2$$

3. Final Solution

$T(n) = \Theta\left(n^1\left(1 + \frac{1}{2}(\log n)^2\right)\right) = \boldsymbol{\Theta(n(\log n)^2)}$

Akra-Bazzi vs. Master Theorem

Scope

- **Master Theorem:** Requires all subproblems to have the same size, $T(n/b)$.
- **Akra-Bazzi:** Allows subproblems of different sizes, $T(n/b_i)$.

Complexity of Use

- **Master Theorem:** Involves a simple polynomial comparison.
- **Akra-Bazzi:** Requires solving for an exponent p and evaluating an integral, which can be more difficult.

Conclusion

Akra-Bazzi is a more general and powerful method, essential for recurrences that do not fit the rigid structure of the Master Theorem.