III. Divide-and-Conquer Recurrences and the Master Theorem

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.
  The *initial conditions* for a sequence specify the terms that precede the first term where the RR takes effect.

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.
  The *initial conditions* for a sequence specify the terms that precede the first term where the RR takes effect.
- Example: Fibonacci numbers: $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with initials $F_0 = 0$ and $F_1 = 1$.

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.
  The *initial conditions* for a sequence specify the terms that precede the first term where the RR takes effect.
- Example: Fibonacci numbers: $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with initials $F_0 = 0$ and $F_1 = 1$.
- A sequence $\{a_n\}$ is called a **solution** of a RR if its terms satisfy the RR.

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.
  The *initial conditions* for a sequence specify the terms that precede the first term where the RR takes effect.
- Example: Fibonacci numbers: $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with initials $F_0 = 0$ and $F_1 = 1$.
- A sequence $\{a_n\}$ is called a **solution** of a RR if its terms satisfy the RR.
- Example:
  - $a_n = 3n$ is a solution of $a_n = 2a_{n-1} - a_{n-2}$ for $n \geq 2$.

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.
  The *initial conditions* for a sequence specify the terms that precede the first term where the RR takes effect.
- Example: Fibonacci numbers: $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with initials $F_0 = 0$ and $F_1 = 1$.

- A sequence $\{a_n\}$ is called a **solution** of a RR if its terms satisfy the RR.
- Example:
  - $a_n = 3n$ is a solution of $a_n = 2a_{n-1} - a_{n-2}$ for $n \geq 2$.
  - In fact, $a_n = 5$ is also a solution. (and $a_n = 2^n$ is not a solution!)

Review: Recurrence relation

- A **recurrence relation (RR)** for the sequence $\{a_n\}$ is an equation that expresses a_n in terms of one or more of the previous terms of the sequence, namely, $a_0, a_1, \ldots, a_{n-1}$, for $n \geq n_0$, where n_0 is a nonnegative integer.
  The *initial conditions* for a sequence specify the terms that precede the first term where the RR takes effect.
- Example: Fibonacci numbers: $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ with initials $F_0 = 0$ and $F_1 = 1$.

- A sequence $\{a_n\}$ is called a **solution** of a RR if its terms satisfy the RR.
- Example:
  - $a_n = 3n$ is a solution of $a_n = 2a_{n-1} - a_{n-2}$ for $n \geq 2$.
  - In fact, $a_n = 5$ is also a solution. (and $a_n = 2^n$ is not a solution!)

- Reading: Handout on "Recurrence Relation Review" (April 2).

Divide-and-Conquer recurrences

- **Divide-and-Conquer (DC) recurrence:**

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

  where

  - constants $a \geq 1$ and $b > 1$,
  - function $f(n)$ is nonnegative, $f(n) \geq 0$.

Divide-and-Conquer recurrences

- **Divide-and-Conquer (DC) recurrence:**

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

  where

  - constants $a \geq 1$ and $b > 1$,
  - function $f(n)$ is nonnegative, $f(n) \geq 0$.

- Example. the cost function of Merge Sort

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + 1 + (n-1)$$

  where

  - $a = 2$ (the number of subproblems)
  - $b = 2$ ($n/2$ is the size of subproblems)
  - $f(n) = 1 + (n-1) = n$ (the cost to divide and combine)

  assuming $n = 2^k$ for $k \geq 1$.

Two methods for finding solutions of DC recurrences

1. Explicit substitution/recursion
2. the master theorem/method

Solving DC recurrences by explicit substitution

- Explicit substitution can be illustrated by the following example

$$T(n) = 4 \cdot T(\frac{n}{2}) + n, \qquad n = 2^k$$

- By iterating the recurrence (i.e. explicit substitution), we have

$$\begin{aligned}
T(n) &= 4 \cdot T(\frac{n}{2}) + n \\
&= 4 \cdot \left(4 \cdot T(\frac{\frac{n}{2}}{2}) + \frac{n}{2}\right) + n = 4^2 \cdot T(\frac{n}{2^2}) + 2n + n \\
&= 4^3 \cdot T(\frac{n}{2^3}) + 2^2 n + 2n + n \\
&= \cdots \\
&= 4^k \cdot T(\frac{n}{2^k}) + 2^{k-1} n + \cdots + 2n + n \\
&= 4^k \cdot T(1) + (2^{k-1} + \cdots + 2 + 1)n \\
&= 4^k \cdot T(1) + \left(\frac{2^k - 1}{2 - 1}\right) n \\
&= n^2 \cdot T(1) + n(n-1) = \Theta(n^2)
\end{aligned}$$

The master theorem/method to solve DC recurrences

- For the DC recurrence, let $n = b^k$, then by recursion[1], we have

$$T(n) = n^{\log_b a} \cdot T(1) + \sum_{j=0}^{k-1} a^j f\left(\frac{n}{b^j}\right)$$

[1] details can be safely skipped for our purpose.

The master theorem/method to solve DC recurrences

- For the DC recurrence, let $n = b^k$, then by recursion[1], we have

$$T(n) = n^{\log_b a} \cdot T(1) + \sum_{j=0}^{k-1} a^j f\left(\frac{n}{b^j}\right)$$

- By carefully analyzing the terms in $T(n)$, we can provide asymptotic bounds on the growth of $T(n)$ in the following **three cases**.

[1] details can be safely skipped for our purpose.

The master theorem/method to solve DC recurrences

Case 1: *If* $n^{\log_b a}$ *is polynomially larger than* $f(n)$, *i.e.,*

$$\frac{n^{\log_b a}}{f(n)} = \Omega(n^{\epsilon}) \quad \textit{for some constant } \epsilon > 0,$$

then

$$T(n) = \Theta(n^{\log_b a}).$$

Example: $T(n) = 7 \cdot T(\frac{n}{2}) + \Theta(n^2)$

The master theorem/method to solve DC recurrences

Case 2: *If* $n^{\log_b a}$ *and* $f(n)$ *are on the same order, i.e.,*

$$f(n) = \Theta(n^{\log_b a}),$$

then

$$T(n) = \Theta(n^{\log_b a} \lg n).$$

Example: $T(n) = 2 \cdot T(\frac{n}{2}) + \Theta(n)$

The master theorem/method to solve DC recurrences

Case 3: *If $f(n)$ is polynomially greater than $n^{\log_b a}$, i.e.,*

$$\frac{f(n)}{n^{\log_b a}} = \Omega(n^{\epsilon}) \quad \textit{for some constant } \epsilon > 0$$

*and $f(n)$ satisfies the regularity condition (*see next slide*), then*

$$T(n) = \Theta(f(n)).$$

Example: $T(n) = 4 \cdot T(\frac{n}{2}) + n^3$

Remarks

1. $f(n)$ satisfies the *regularity condition* if

$$a \cdot f\left(\frac{n}{b}\right) \leq cf(n)$$

   for some constant $c < 1$ and for all sufficient large n.
2. The proof of the master theorem is involved, shown in section 4.6, which we can safely skip.
3. The master theorem doesn't cover all possible cases, and the master method cannot solve every DC recurrences.