PAGE NO.
DATE / /

AOA week 5 Notes

Divide and conquer:

Divide into subproblems

Conquer i.e solve the subproblems recursively, or if trivial solve the problem itself.

Combine the solution to the subproblems.

Eg: merge-sort (A, p, r)
    if p<r then
        Q = ⌊(p+r)/2⌋ → Divide $O(1)$
        merge-sort (A, p, Q)
        merge-sort (A, Q+1, r) } ⇒ conquer $2T(n/2)$
        merge (A, p, Q, r) } ⇒ combine $O(n)$.

$$T(n) = \begin{cases} 1 & \text{if } n==1 \\ \underbrace{2T(n/2)}_{\text{conquer}} + \underbrace{O(n)}_{\text{merge}} + \underbrace{O(1)}_{\text{Divide}} \end{cases}$$

General recurrence equation.

$$T(n) = \begin{cases} \Theta(1) & n==1 \\ \underbrace{a}_{\text{no. of subproblems}}T(\underbrace{n/b}_{\text{size of subproblem}}) + \underbrace{D(n)}_{\text{Time to divide}} + \underbrace{C(n)}_{\text{Time to conquer.}} \end{cases}$$

Master Method

$$T(n) = aT(n/b) + f(n)$$

$a \geq 1$ $b \geq 1$ are constants

1) IF $F(n) = O(n^{\log_b a - \varepsilon})$ For some $\varepsilon > 0$
$T(n) = \Theta(n^{\log_b a})$

2) IF $f(n) = \Theta(n^{\log_b a})$ then
$T(n) = \Theta(n^{\log_b a} \log n)$

3) IF $F(n) = \Omega(n^{\log_b a + \varepsilon})$ For some $\varepsilon > 0$
and if $a f(n/b) \leq c \cdot f(n)$ For some constant $c < 1$

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

Generalization
$f(n) = \Theta(n^{\log_b a} \log^k n)$
where $k \geq 0$.

Then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$

PAGE No:
DATE / /

eg: merge sort

$$T(n) = 2T(n/2) + O(1) + O(n)$$

$$= 2T(n/2) + O(n)$$

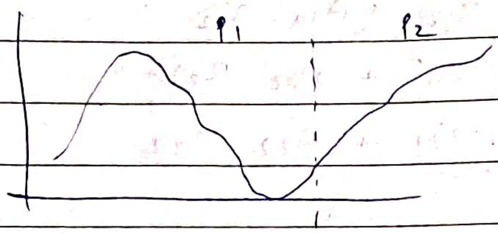
$$n^{\log_b a} = n^{\log_2 2} = n^1$$

$$f(n) = n$$

$\therefore$ case # 2

$$O(n \log n)$$

eg: stock market problem.



case 1 : sell & Buy in P_1

Case 2 : sell & Buy in P_2

case 3 : sell in P_2 , Buy in P_1

case 3

$$M = \text{Min}(M_1, M_2)$$

$$X = \text{Max}(X_1, X_2)$$

$$B = M_1$$

$$S = X_2$$

$$f(n) = C(n) + D(n)$$

$$= O(1) + O(1)$$

$$= O(1)$$

$$n^{\log_b a} = n^{\log_2 2} = n^1$$

case #1 $T = \Theta(n)$

eg: Dense Matrix Multiplication

$$n\left\{\underbrace{[A]}_{n}\right. \underbrace{[B]}_{n}\left.\right\}n = \underbrace{[C]}_{n}\left.\right\}n$$

Brute Force $\Rightarrow \Theta(n^3)$

$$\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right] = \left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right]$$

$C_{11} = A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$ → subproblems

$C_{12} = A_{11} \cdot B_{12} + A_{12} \cdot B_{22}$

$C_{21} = A_{21} \cdot B_{11} + A_{22} \cdot B_{21}$

$C_{22} = A_{21} \cdot B_{21} + A_{22} \cdot B_{22}$

→ conquer.

$$f(n) = D(n) + C(n)$$
$$= O(1) + \Theta(n^2) = \Theta(n^2)$$

$$n^{\log_b a} = n^{\log_2 8} = n^3$$

case #1 $\Theta(n^3)$

suppose initially $n=4$.

$$\left[\begin{array}{c|c} 2\times2 & 2\times2 \\ \hline 2\times2 & 2\times2 \end{array}\right]$$

2×2 matrix

8 multiplication

∴ subproblems = 8

size reduced by 2

Strassen's Matrix Multiplication

$P = (A_{11} + A_{22})(B_{11} + B_{22})$

$Q = (A_{21} + A_{22})B_{11}$

$R = A_{11}(B_{12} - B_{22})$

$S = A_{22}(B_{21} - B_{11})$

$T = (A_{11} + A_{12})B_{22}$

$U = (A_{21} - A_{11})(B_{11} + B_{12})$

$V = (A_{12} - A_{22})(B_{21} + B_{22})$

$C_{11} = P + S - T + V$

$C_{12} = R + T$

$C_{21} = Q + S$

$C_{22} = P + R - Q + U$

$f(n) = \Theta(1) + \Theta(n^2) = \Theta(n^2)$

$n^{\log_b a} = n^{\log_2 7} = n^{2.81}$

Case #1 $\Theta(n^{2.81})$

pg 17 * Finding Min Max in unsorted array

pg 18 * closest pair of points problem (2D)