Lecture 5. Divide & Conquer

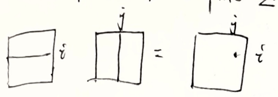
Dec. 14, 2020

$T(n) \leq a \cdot T(\frac{n}{b}) + c \cdot n^d$

$$T(n) \leq \begin{cases} O(n^{\log_b a}) & \log_b a > d \\ O(n^d) & \log_b a < d \\ O(n^d \log n) & \log_b a = d \end{cases}$$

1. Matrix Multiplication

X, Y $(n \times n)$ compute $Z = XY$



X Y Z

$\Theta(n^3)$, each ~~$\Theta(n)$~~

$\Theta(n)$, n^2 elements

$$\underset{X}{\begin{array}{cc|l} n/2 & n/2 & \\ A & B & n/2 \\ C & D & n/2 \end{array}} \quad \underset{Y}{\begin{bmatrix} E & F \\ G & H \end{bmatrix}} = \underset{Z}{\begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}}$$

$T(n) \leq 8T(\frac{n}{2}) + \underbrace{C \cdot n^2}_{\text{Addition costs } \Theta(n^2)}$

Master Theorem: $T(n) \leq aT(\frac{n}{b}) + c \cdot n^d$ $\begin{cases} a = 8 \\ b = 2 \\ d = 2 \end{cases}$

$\log_b a = 3 > d = 2$

$T(n) = O(n^{\log_b a}) = O(n^3)$

Strassen Multiplication (1969)

$P_1 = A \times (F - H)$ $P_2 = (A+B)H$ $P_3 = (C+D)E$ $P_4 = \sim \ldots$

$P_7 = (A-C)(E+F)$

$T(n) \leq 7T(\frac{n}{2}) + c \cdot n^2$ $T(n) = O(n^{\log_2 7}) = O(n^{2.807\ldots})$

Coppersmith - Winograd (1987) $T(n) = O(n^{2.3754})$

Stothers (2010) $T(n) = O(n^{2.3736})$

Vassilevka - Williams (2012) $T(n) = O(n^{2.372873})$

LeGall (2014) $T(n) = O(n^{2.3728639})$

2. Sorting (Merge Sort)

$a = 2$

MergeSort $(A[1\cdots n])$: $\quad T(n) \leq 2\cdot T(\frac{n}{2}) + Cn \quad b=2$

$B = MS(A[1\cdots \frac{n}{2}])$ $\quad \log_b a = 1 = d \quad d=1$

$C = MS(A[\frac{n}{2}+1 \cdots n])$

return Merge (B, C) $\quad T(n) = O(n\log n)$

queue $Q = [A_1\ A_2\ A_3\ A_4 \ldots A_n]$

pop two elements, mergesort and add it

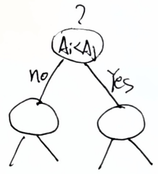
Han (2002) $\quad O(n \log\log n)$ $\quad$ deterministic algorithms

Han - Thorup (2002) $\quad O(n\sqrt{\log\log n})$ $\quad$ randomized algorithm

Runtime is variable, $O(f(n))$ is expected run-time

The fastest algorithm based comparison is $\Theta(n\log n)$ (worst-case)

Input: Some unknown permutation σ of $1, 2, \ldots, n$



depth T: number of comparisions

leaves : $2^T \geq n!$ Every leaf corresponds to a unique σ

$T \geq \log_2 n!$

$\geq n\ln n - O(n)$

$= \Omega(n\log n)$

Median/Select $\quad A[1\cdots n] \quad 1 \leq k \leq n$

Goal: output kth smallest of A. (median is $k = n/2$)

$\boxed{\text{QuickSelect}}$ $\quad [\,0\ 0\ 0 \rightarrow 0 \qquad 0\,]$

pivot (107th smallest)

$\begin{cases} k = 107 & \text{return pivot} \\ k < 107 & \text{left} \\ k > 107 & \text{right} \end{cases}$

Desire (median): $T(n) \leq T(\frac{n}{2}) + c\cdot n \quad O(n)$

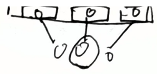
bad case (smallest or largest): $T(n) \leq T(n-1) + c\cdot n \quad O(n^2)$

A has no duplicate entries. $B = [(A_1, 1), (A_2, 2), \cdots (A_n, n)]$

comparison for B: first compare 1st, if equal compare 2nd

$\boxed{\text{Select}(A[1\cdots n], k)}$

(1) Break A into groups of size 5 each



(2) B = array of medians in each group

(3) p = select $(B, \frac{n}{10})$ $\quad \frac{n}{10}$ means $\frac{n}{5} \cdot \frac{1}{2}$ ~~find~~ median.

(4) $L = \{<p\}$, $R = \{>p\}$ $\quad P > \frac{1}{2} \cdot \frac{n}{5} \cdot 3 = \frac{3n}{10}$

bigger than half, each group bigger than 2, 3.

(5) if $K = |L|+1$, return p

elif $K \leq |L|$, return select (L, K)

else return select $(R, K-|L|-1)$

$T(n) \leq O(n) + \frac{n}{5} \cdot O(1) + T(\frac{n}{5}) + O(n) + T(\frac{7n}{10})$

$\leq \underbrace{T(\frac{n}{5})}_{\text{select to find pivot}} + \underbrace{T(\frac{7n}{10})}_{\text{pivot's left / right (based on k)}} + \underline{C \cdot n}$ comparision

In facts divide into X ($X \geq 5$, odd) parts are all right.

(If $X=3$, doesn't work)

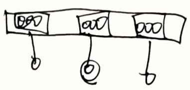
$\boxed{\text{Guess } T(n) \leq B \cdot n}$ (Note: B is same for all n)

Induction: Base Case as long as $B \geq 1$, true

Inducative step: $B \cdot \frac{n}{5} + B \cdot \frac{7n}{10} + C \cdot n \leq B \cdot n$

$B \geq 10C$

if divide into groups of size 3:



~~$T(n) \leq T(\frac{1}{2} \cdot \frac{n}{3} \cdot 2) + T(\frac{n}{3}) +$~~

$T(n) \leq T(n - \underbrace{\frac{1}{2} \cdot \frac{n}{3} \cdot 2}_{\text{guarantee to bigger than this}}) + T(\frac{n}{3}) + C \cdot n$

$\leq T(\frac{2n}{3}) + T(\frac{n}{3}) + C \cdot n$

Induction: $T(n) \leq B \cdot n$ Base case: true

$T(n) \leq \frac{2}{3} n \cdot B + \frac{n}{3} \cdot Bn + C \cdot n \leq B \cdot n$

$0 \geq C$ (false)

~~So $T(n) = O(n)$~~ so $T(n) > B \cdot n$

If we wants the induction be true, divide into X parts.

then. $B(1 - \frac{1}{2} \cdot \frac{1}{X} \cdot \frac{X+1}{2})n + B\frac{n}{X} \cdot n + C \cdot n \leq B \cdot n$

$B(1 - \frac{1}{X} - 1 + \frac{X+1}{4X}) \geq C$

$B \cdot \frac{X-3}{4X} \geq C$

$B \geq \frac{4CX}{X-3}$, so $X > 3$