CS 170

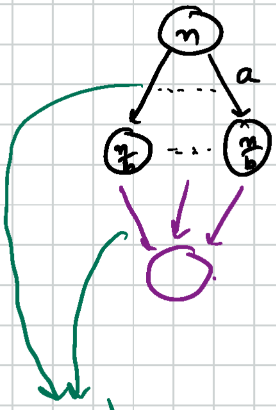
Lecture 3

- use iClicker

Sanjam Garg.

join.iclicker.com/RGWR

Divide & Conquer



- Binary Search
- Sorting
- Integer multiplication
- Matrix multiplication
- Median finding.

$$T(n) = a\,T\left(\frac{n}{b}\right) + O(n^d)$$

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

Median Finding

Input: A list of n numbers (unsorted)

Output: Find the $\lceil \frac{n}{2} \rceil^{th}$ smallest number in the list

Select

Input: unsorted list $S = \{a_1 \dots a_n\}$, integer $k \in \{1 \dots n\}$

Output: k^{th} smallest number in S

* Median $(S) \rightarrow$ Select$(S, \lceil \frac{n}{2} \rceil)$

Algorithm: Select (S, k)

* Pivot $v \leftarrow S = \{a_1 \dots a_n\}$

* Split $S \rightarrow S_L \quad S_v \quad S_R$

$$S_L = \{a \in S \mid a < v\}$$

$$S_v = \{a \in S \mid a = v\}$$

$$S_R = \{a \in S \mid a > v\}$$

* case analysis

$k \leq |S_L| \rightarrow$ Select(S_L, k)

$|S_L| < k \leq |S_L| + |S_v| \rightarrow$ Return v

$|S_L| + |S_v| < k \rightarrow$ Select$(S_R, k - |S_L| - |S_v|)$

Goodcase: $|S_L| \approx |S_R|$

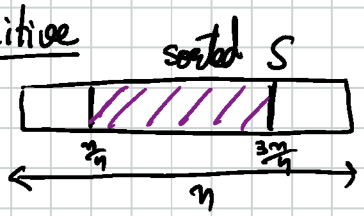
Bad case: $|S_L|$ or $|S_R| = 1$

$\hookrightarrow O(n^2)$

$T(n) =$ Expected number of steps taken.

$$= \sum_t t \, p_t$$

Intuitive



$$T(n) = T\left(\frac{3n}{4}\right) + O(n)$$

$$= O(n)$$

Good pivot: v is a good pivot if it is between $\frac{n}{4}^{th}$ smallest element and the $\frac{3n}{4}^{th}$ smallest element.

observation 1: if v is a good pivot then

$$\max\{|S_L|, |S_R|\} \le \frac{3n}{4}$$

also, call to select is on a list of size $\le \frac{3n}{4}$

observation 2: $\#$ of good pivots $= \frac{n}{2}$

$$T(n) = E(\#\text{ of steps to compute Select on input } S \text{ of size } n)$$

$$= E\left(\#\text{ of steps to reduce list to size } \le \frac{3n}{4}\right)$$

$$+ \underbrace{E\left(\#\text{ of steps to compute Select on input list of size} \le \frac{3n}{4}\right)}_{\le T\left(\frac{3n}{4}\right)}$$

$$\le E(\#\text{ of pivots to get a good pivot})^{*}\, cn \quad + \quad T\left(\frac{3n}{4}\right)$$

$$E = 1 \times \frac{1}{2} + 2 \times \frac{1}{2} \times \frac{1}{2} + 3 \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \cdots\cdots$$

$$- \quad \frac{E}{2} = \qquad 1 \times \frac{1}{2} \times \frac{1}{2} + 2 \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} \cdots$$

$$\frac{E}{2} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} \cdots\cdots$$

$$= \frac{1}{2} \times \frac{1}{1 - \frac{1}{2}}$$

$$\Rightarrow \quad E = 2$$

$$T(n) \le 2cn + T\left(\frac{3n}{4}\right) \longrightarrow T(n) = O(n)$$

Algorithm (Deterministic) → T(n)

* Pivot $v \leftarrow$ Pick Pivot (S) ~~$S = \{a_1 \ldots a_n\}$~~

* Split $S \longrightarrow S_L \quad S_v \quad S_R$

$$S_L = \{a \in S \mid a < v\}$$

$$S_v = \{a \in S \mid a = v\}$$

$$S_R = \{a \in S \mid a > v\}$$

* case analysis

$k \le |S_L| \rightarrow \underbrace{\text{Select}}_{②a}(S_L, k)$ $\le T\left(\frac{7n}{10}\right)$

$|S_L| < k \le |S_L| + |S_v| \rightarrow$ Return v

$|S_L| + |S_v| < k \rightarrow \underbrace{\text{Select}}_{②b}(S_R, k - |S_L| - |S_v|)$

Pick Pivot (S) ↪ $\{a_1 \ldots a_n\}$

* Split S

cn $\quad \underbrace{\{a_1 \ldots a_5\}}_{1}, \underbrace{\{a_6 \ldots a_b\}}_{2} \ldots\ldots \frac{n}{5}$

* Median of each list

$c'n$ $\quad b_1 \quad b_2 \quad - - \quad b_{n/5}$

* Median in $\{b_1 \ldots b_{\frac{n}{5}}\}$

$T\left(\frac{n}{5}\right)$ $\underbrace{\text{Select}}_{①}\left(\{b_1 \ldots b_{\frac{n}{5}}\}, \left\lceil \frac{n}{10} \right\rceil\right)$

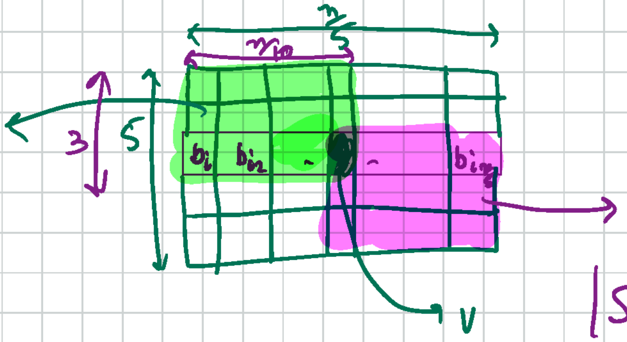
Median of medians

all $\le v$

⇓

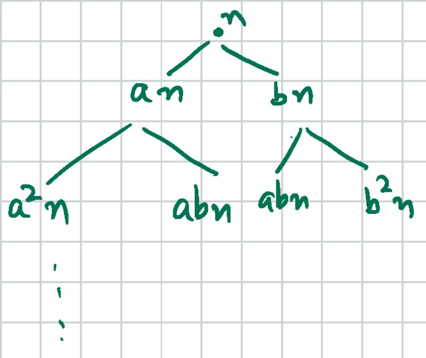
$$|S_R| \le n - \frac{3n}{10}$$

$$= \frac{7n}{10}$$



all $\ge$ V

$$|S_L| \le n - \frac{3n}{10} = \frac{7n}{10}$$



$$\underset{cn}{T(n)} \le T\left(\frac{7n}{10}\right) + T\left(\frac{n}{5}\right) + O(n)$$

$$(a+b)cn \quad = \quad T(an) + T(bn) + O(n)$$

$$(a+b)^2 cn \quad = \quad cn + (a+b)cn + (a+b)^2 cn \ldots$$

$$(a+b)^3 cn \quad = \quad O(n) \quad \text{if} \quad a+b < 1$$

$\vdots$

Exponentiation

Input: Number n

Output: 2^n in decimal

Naive: $\underbrace{2 \times 2 \times 2 \cdots 2}_{n \text{ times.}} \longrightarrow n$ multiplications.

↳ $O(n \underbrace{M(n)})$

↓

# of steps to multiply a n digit number.

$2^{100} \rightarrow (2^{50})^2$

$2^{50} \rightarrow (2^{25})^2$

$2^{25} \rightarrow (2^{12})^2 \times 2$

$2^{12} \rightarrow (2^{6})^2$

$2^{6} \rightarrow (2^{3})^2$

$2^{3} \rightarrow (2^{1})^2 \times 2$

$2^{1} \rightarrow 2$

```
        Exp(a, n):                              A
A           if n = 1  return a
          B ← b = Exp(a, $\lfloor n/2 \rfloor$)
A           if n is even  return   B   B
                                   b * b
            if n is odd   return   B   B
                                   b * b  a  → A
```

→ # of multiplications $\log n$

$$T(n) = T\left(\frac{n}{2}\right) + O(\underbrace{M(n)}_{\geq n})$$

$$= O(M(n))$$

Fibonacci Sequence: 0, 1, 1, 2, 3, 5, 8 . . . . . .

$\forall n$

$2F_{n-1} \geq F_n \geq 2F_{n-2}$

$\downarrow$

$2^{\frac{n}{2}} \ldots 2^n$

$\downarrow$

$2^{0.6n}$

$$F_n = \begin{cases} 0 & \text{if } n=0 \\ 1 & \text{if } n=1 \\ F_{n-1}+F_{n-2} & \text{otherwise} \end{cases}$$

```
Fib(n):
        if n=0      return 0
        if n=1      return 1
        return      Fib(n-1) + Fib(n-2)
```

$\downarrow$

exponential

```
Fib(n)
    if   n=0          return 0
    create an array   F[0... n]
    F(0)=0            F(1)=1
    for  i= 2 .... n
         F(i) = F(i-1) + F(i-2)
    return F(n)
```

$1 + 2 + 3 \ldots n \rightarrow O(n^2)$

$\underline{O(n^2)}$

$$\begin{bmatrix} F_2 \\ F_1 \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}^{A} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$\begin{bmatrix} F_3 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_2 \\ F_1 \end{bmatrix}$$

$\vdots$

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix}$$

$O(\log n)$

$$\begin{bmatrix} F_{n+1} \\ F_n \end{bmatrix} = A^n \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} = A^n \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

```
Fib(n):
        if $n=0$    return 0
        $\begin{bmatrix} F_{n+1} \\ f_n \end{bmatrix} = Exp(A, n) \times \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
        return $F_n$
```

$$T(n) = T\left(\frac{n}{2}\right) + O(M(n))$$

$$= O(M(n)) \rightarrow \quad O\left(n^{\log_2 3}\right)$$