Randomized Algo

def: algo(k) returns $1 \le i \le k$ in $O(1)$

Why randomized?

- can be simpler.
- can be faster.

Models:

- deterministic algo f

  worst case input x, $x \mapsto f(x)$

  complexity: $\max_{|x|=n} T(x)$

def Ω finite / countable infinite set

$Pr : \Omega \to [0,1]$, $\left(\sum_{w \in \Omega} Pr[w]\right) = 1$

$E \subseteq \Omega$, $Pr(E) = \sum_{w \in E} Pr[w]$

random variable : function $X : \Omega \to \mathbb{R}$

$\downarrow$ has

expectation : $\mathbb{E}[X] = \sum_{w \in \Omega} X(w) Pr[w]$

Model : probablistic

deterministic algo f

$x \xleftarrow{\$} \mathcal{D}_n$ $\leftarrow$ distribution over size n input.

$x \mapsto f(x)$

complexity

$T(n) = \mathbb{E}_{x \leftarrow \mathcal{D}} [T(x)]$

rmk: theory is brittle.

Model : randomized algo :

f is randomized.

worst case input x

$x \mapsto f(x)$

$\Rightarrow$ outputs a RV

complexity

$$\underbrace{\max_{|x|=n} \mathbb{E}[T(x)]}_{T(n)}$$

Analysis :

correctness : outputs correct answer always.

complexity : expected run time.

Communication

n People $P_1, \ldots, P_n$. In each round j, each P_i tries to communicate.

def $X_{ij} = \begin{cases} 1 & P_i \text{ attempts to communicate at round } j. \\ 0 & \end{cases}$

independent

$$\Pr[X_{ij} = \underbrace{a}_{\{0,1\}} \wedge X_{i'j'} = \underbrace{b}_{\{0,1\}}] = \Pr[X_{ij} = a] \cdot \Pr[X_{ij} = b]$$

$$\mathbb{E}[X_{ij}] = 1 \cdot \Pr[X_{ij} = 1] + 0 \cdot \Pr[X_{ij} = 1]$$
$$= \Pr[X_{ij} = 1] = P$$

$\mathbb{E}[\#\text{people communicating}]$

$$= \mathbb{E}[\sum_i X_{ij}] = \sum_i \mathbb{E}[X_{ij}] = n \cdot P$$

lem $1+x \le e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

cor $\left(1-\frac{1}{n}\right)^{n-1} \ge \frac{1}{e}, \quad n > 1$

$$\frac{1}{\left(1-\frac{1}{n}\right)^{n-1}} = \left(\frac{n}{n-1}\right)^{n-1} = \left(1+\frac{1}{n-1}\right)^{n-1} \le \left(e^{\frac{1}{n-1}}\right)^{n-1} \le e$$

def Bernouli random variables. w/ parameter p

$X \in \{0,1\}$, $\Pr[X=1] = p$, $\Pr[X=0] = 1-p$

$X_1, \dots, X_n$ independent bernouli

A geometric RV w/ param p is $Y = \min\{i : X_i = 1\} \in \mathbb{N}$

lem • $Y \geq 1$

• $\Pr[Y \geq i] = \underbrace{\Pr[X_1, X_2, \dots, X_{i-1} = 0]}_{\text{all fails}} = (1-p)^{i-1} \Rightarrow$ independent

• $\Pr[Y = i] = \Pr[\ \text{——}\ {''}\ \text{——}\ , X_i = 1] = (1-p)^{i-1} \cdot p$

lem Z a RV over $\mathbb{N}$. Following equiv

(a) $\mathbb{E}[Z] = \sum_{i=0}^{\infty} i \cdot \Pr[Z = i]$

(b) $\mathbb{E}[Z] = \sum_{i=0}^{\infty} \Pr[Z \geq i]$

Cor $\mathbb{E}[\text{Geom}(p)] = \frac{1}{p}$

$= \sum_{i \geq 1} [\text{Geom}(p) \geq i]$

$= \sum (1-p)^{i-1} = 1 + (1-p) + (1-p)^2 + \cdots$

$= \frac{1}{1-(1-p)} = \frac{1}{p}$

$p \to 0 \Rightarrow \mathbb{E} \to \infty$

def

$a = (a_1, \dots, a_n)$ distinct integers.

$\text{rank}(a_i) = |\{ j : a_j < a_i \}| + 1$ $\Rightarrow$ position in sorted array.

def given $a = (a_1, \dots, a_n)$, $1 \le m \le n$,

the $\boxed{\text{selection problem}}$ is to output a_i s.t. $\text{rank}(a_i)$

prop selection is in $O(n \log n)$ deterministic time.

- sort a in $O(n \log n)$ time
- output m^{th} element

$\Rightarrow$ comparison sort takes $\Omega(n \log n)$.

fact selection can be done deterministically $O(n)$ time.

MOM select. $\Rightarrow$ poor constants.

Thm: Selection in $O(n)$ expected time.

Algo: some select (a, m)

- use some rule to pick spliter a_i
- write $a = b \circ a_i \circ c$, $b_j < a_i < c_k$ $\Rightarrow$ rank $(a_i) = |b| + 1$
- if $m = |b| + 1$, return a_i
  if $m < |b| + 1$, return some-select (b, m)
  if $m > |b| + 1$, return some-select $(c, m - (|b| + 1))$

prop. any splitter, this algo is correct.

prop Suppose pick splitter in $O(n)$, then some-select in $O(n^2)$.

pf:

$$\underset{\displaystyle \max_i T(a_i)}{\overset{\displaystyle T(a)}{\|}} \le \underbrace{O(n)}_{\text{splitter}} + \underbrace{O(n)}_{\text{split}} + \underbrace{\max \{ T(b) + T(c) \}}_{\text{recursion}}$$

$$\le O(n) + T(n-1)$$

$$\le O(n^2)$$

idea: pick balanced split.
$\hookrightarrow$ rank $(a_i) \in [\frac{n}{4}, \frac{3}{4}n]$ $\to$ achieving this is a selection problem

Notice: half a_i in the range. $\Rightarrow$ choose a_i randomly.