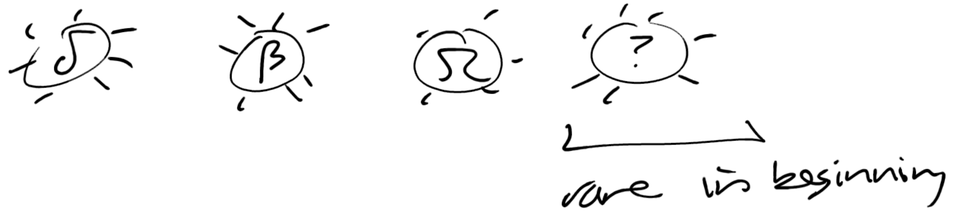
CS473 Algorithms: Lecture 13 (2022-03-03)

logistics: pset 5 out #17

last lecture: - randomized algo - motivation
- def
- concentration reduction

today: randomized algo

Q: detecting next covid variant?



rare in beginning

def: A bernoulli random variable w/ parameter p

is a rand variable X over $\{0,1\}$

$\Pr[X=1] = p$

$\Pr[X=0] = 1-p$

def: $X_1, X_2, \ldots$, bernoulli param p

√ independent

A geometric rand var param p

is $Y = \min\{i : X_i = 1\} \in \mathbb{N}$

ex: - $Y \geq 1$

- $\Pr[Y=i]$

$= \Pr[X_i = 1 \wedge X_1 = \cdots = X_{i-1} = 0]$

$= p(1-p)^{i-1}$

- $\Pr[Y \geq i]$

$= \Pr[X_1 = \cdots = X_{i-1} = 0]$

$= (1-p)^{i-1}$

Lem. Z rand var over $\mathbb{N}$

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

(b) $\mathbb{E}[Z] = \sum_{i=1}^{\infty} \Pr[Z \geq i]$

pf: (a), for countable Ω:

$$\mathbb{E}[Z] = \sum_{\omega \in \Omega} Z(\omega) \cdot \Pr[Z=\omega]$$

$$= \sum_{i=0}^{\infty} i \cdot \underbrace{\sum_{\omega : Z(\omega)=i} \Pr[Z=\omega]}_{\Pr[Z=i]}$$

(b):

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

$$= \sum_{i=0}^{\infty} \sum_{1 \leq j \leq i} \Pr[Z=i]$$

$$= \sum_{1 \leq j \leq \infty} \underbrace{\sum_{i \geq j} \Pr[Z=i]}_{\Pr[Z \geq j]}$$

□

Cor. $\mathbb{E}[\mathrm{Geo}(p)] = 1/p$

pf:

$$= \sum_{i \geq 1} \underbrace{\Pr[\mathrm{Geo}(p) \geq i]}_{(1-p)^{i-1}}$$

$$= \underbrace{(1-p)^0}_{=1} + (1-p) + (1-p)^2 + \cdots$$

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

□

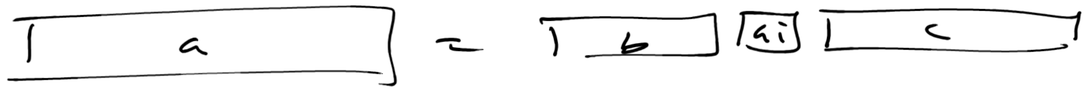
rmk. $1/p \xrightarrow[p \to 0]{} \infty$

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

$$\text{rank}(a_i) = \{ \#j : a_j < a_i \} + 1$$

Lem: $\text{rank}(a_i)$ is position of a_i in sorted (increasing) order of a

eg:



$$b_j < a_i < c_k \quad \forall j,k$$

$$\text{rank}(a_i) = |b| + 1$$

def: given $a = (a_1, \dots, a_n)$ $1 \le m \le n$, the selection problem is to output a_i s.t. $\text{rank}(a_i) = m$. the median problem is $m = \lceil n/2 \rceil$

prop: selection in deterministic $O(n \lg n)$ time

sketch: sort a in $O(n \lg n)$ time

output m-th element in the sorted order ☐

fact: (comparison) sorting requires $\Omega(n \lg n)$ time

fact [cs374]: selection can be done in $O(n)$ deterministic time

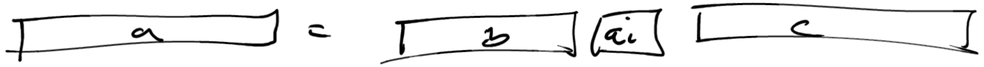
Thm: not "simple" par constants

thm: selection in $O(n)$ expected time

algo - some-select(a, m)

- use [some] rule to pick splitter a_i
- write $a = b, a_i, c$

$$b_j < a_i < c_k \quad \forall j,k$$



$\hookrightarrow$ $rank(a_i) = |b| + 1$

- 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] method to pick splitter is correct

prop: suppose can pick splitter in $O(n)$ time deterministically
then some-select runs in $O(n^2)$ time

pf: $T(n) = \underbrace{T(a)}_{= \max_{|a| \le n} T(a,m)} \le \underbrace{O(n)}_{\text{splitter}} + \underbrace{O(n)}_{\text{split}}$

$$+ \max\{T(b), T(c)\}$$

$|a| = |b| + 1 + |c|$

$\hookrightarrow$ $|b|, |c| < n$

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

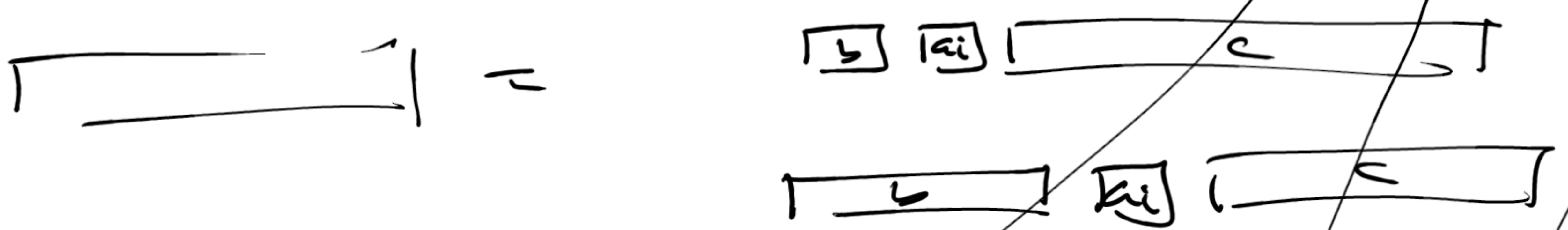
$= \max_{|a| \le n} T(a)$

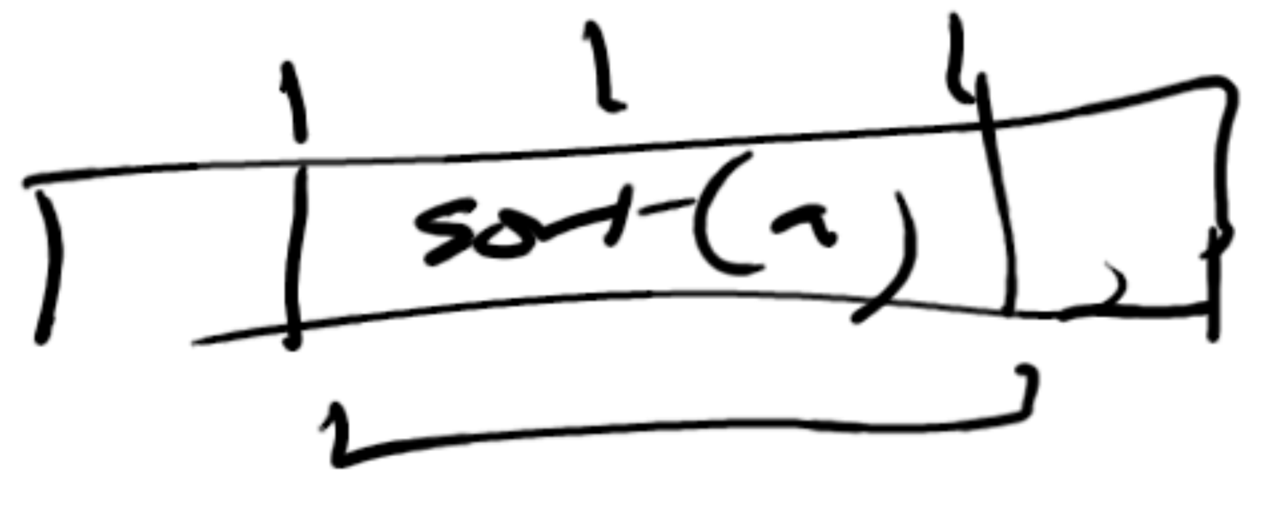
$\le \dots$

$\le O(n^2)$ □

rmk: [worse] than sorting

idea: pick splitter where $\text{rank}(a_i) \in [n/4, 3n/4]$





[bad] is a selection problem!

idea: [any] a_i ✓ suffices

[half] of a_i are in

⇒ pick a_i randomly

algo: pick-splitter(a)

while

$O(1)$ – pick $i \in [1, n]$ randomly

$O(n)$ – write $a = b, a_i, c$

$b_j < a_i < c_k \quad \forall j, k$

$O(n)$ – if $\text{rank}(a_i) \in [n/4, 3n/4]$

$|b|+1$

return a_i

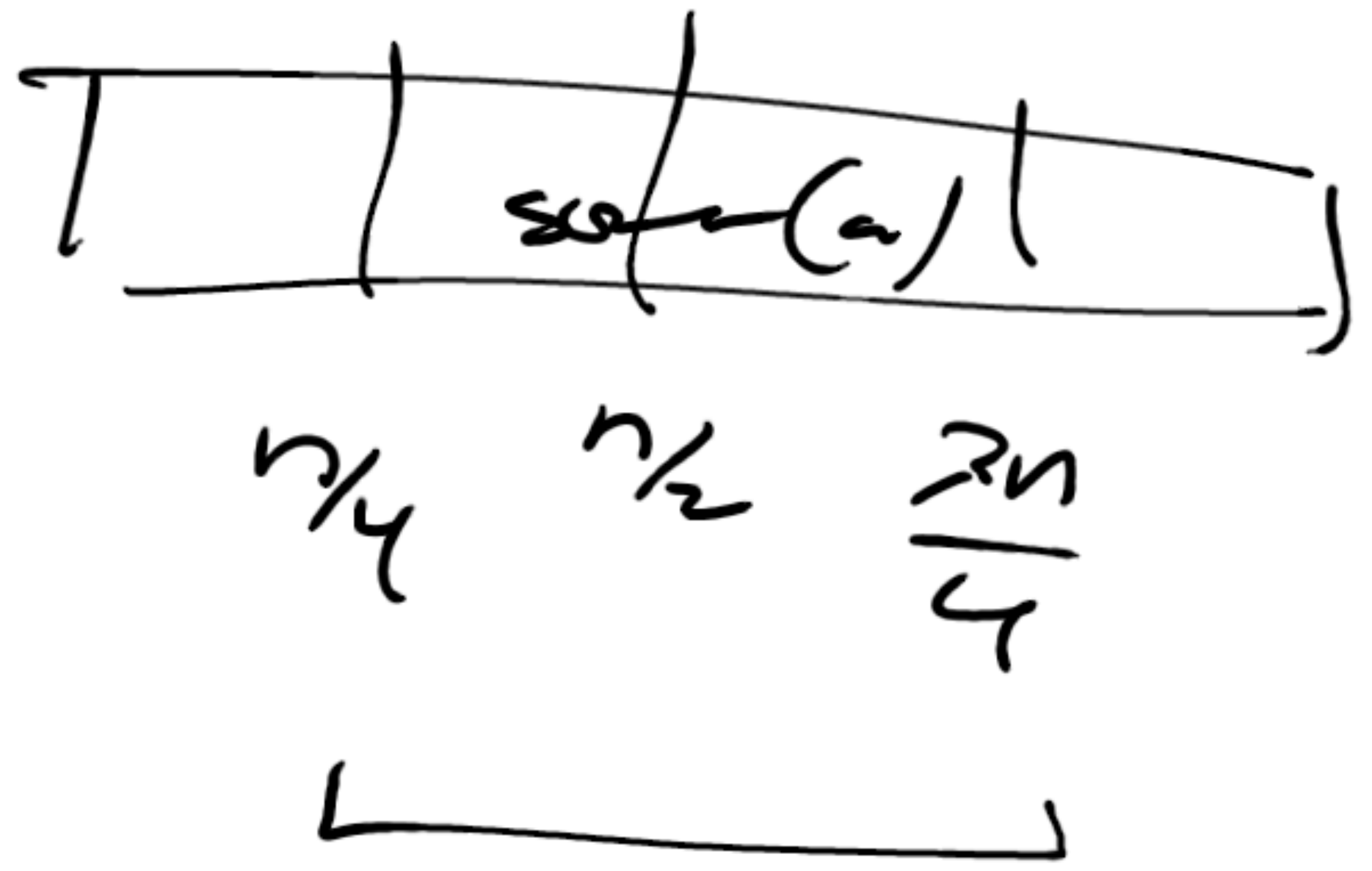
lem: pick-splitter has $O(n)$ expected runtime

pf: Let L be # loop iterations

clm: runtime $\in O(n \cdot L)$

clm: $L \sim \text{Geom}(1/2)$

pf: $\Pr[\text{rank}(a_i) \in [n/4, 3n/4]] = \frac{1}{2}$



$\Rightarrow \mathbb{E}[\text{runtime}] = O(n) \cdot \underbrace{\mathbb{E}[L]}_{=2}$

∎

cor: some-select w/ pick-splitter ch. hs

$O(n)$ expected runtime

pf:

$$T(a,m) \le \underbrace{P(a)}_{\text{pick splitter}} + \underbrace{O(n)}_{\text{split } a \text{ into } b, a_0, c} + \max\{T(b,m), T(c, m-(|b|+1))\}$$

linearity of expectation

$$\mathbb{E}[T(a,m)] \le \underbrace{\mathbb{E}[P(a)]}_{O(n)} + O(n) + \max\{\mathbb{E}[T(b,m)], \mathbb{E}[T(c,m-(|b|+1))]\}$$

$\frac{n}{4} \le |b|, |c| \le \frac{3n}{4}$

$|a| = |b| + 2 + |c| \Rightarrow$

$rank(a_i) = |b|+1$

$\in [n/4, 3n/4]$

$\le T(\frac{3n}{4})$

"

$\max\limits_{\substack{|d| \le \frac{3n}{4} \\ 1 \le l \le |d|}} \mathbb{E}[T(d,l)]$

$$T(n) \le O(n) + O(n) + T(3n/4)$$
$$\le \ldots$$
$$\le O(n)$$

cor: can compute median in $O(n)$ expected time

rmk: "simple" algorithm, less analysis, good constants

Q: selection vs sorting?

↳ sorting entire list

↳ "sorting one element"

also: quicksort(a)

- use some rule to pick splitter a_i
- write $a = b, a_i, c$

$b_j < a_i < c_k \quad \forall j,k$

- return quicksort(b), a_i, quicksort(c)

prop: quicksort always sorts correctly

rmk: suppose choosing splitter takes $O(n)$ deterministically then quicksort runs in $O(n^2)$ time deterministically

pf: $T(a) = \underbrace{O(n)}_{\text{pick splitter}} + \underbrace{O(n)}_{\text{split}} + T(b) + T(c)$

$|a| = |b| + 1 + |c|$

$\text{rank}(a_i) - |b| + 1 = m$

$T(n) \leq O(n) + T(m-1) + T(n-m)$

$\leq \cdots$

$\leq O(n^2)$

rmk - [is] $O(n^2)$ at $m=1$ always

$\Rightarrow T(n) = \Theta(n) + T(n-1)$

$\leq \cdots = \Theta(n^2)$

idea: split | evenly

↳ [at] median

↖ select$(a, n/2)$

Lem: quicksort w/ splitter as median in $O(n)$ expected time

takes $O(n \lg n)$ expected time

pf: $T(n) \leq \underbrace{M(n)}_{\text{median}} + \underbrace{O(n)}_{\text{split}} + T(|L|) + T(|G|)$

$$\mathbb{E}[T(n)] \leq \underbrace{\mathbb{E}[M(n)]}_{O(n)} + O(n) + \mathbb{E}[T(|L|)] + \mathbb{E}[T(|G|)]$$

$|L| = |G| = n/2$

$$\leq \max_{|d| \leq n/2} \mathbb{E}[T(d)] = T(n/2)$$

$$T(n) \leq O(n) + 2 \cdot T(n/2)$$

$$\leq \dots$$

$$\leq O(n \lg n)$$ ∎

rmk: achieves optimal $\Theta(n \lg n)$ runtime of comparison sort

today:

- rand also
  - geometric rand vars
  - randomized selection
  - randomized quicksort

next lecture: rand ds's

Logistics: pset 5 FAQ