Michael A. Forbes
miforbes@illinois.edu
2024-02-29

CS473 Algorithms: Lecture 13 (2024-02-29)

logistics: pset 5 due Fri

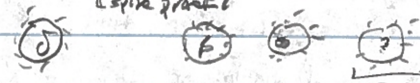
last lecture: randomized algorithms
- ~~[illegible]~~
- def
- contention resolution

today: rand algs

reading: KT 13.3, 13.5

Q – detecting next covid variants?

[spike protein]



[how to model?] rare in beginning [how long to find it?] [only small # samples tested]

def: a bernoulli random variable w/ parameter p is a random variable X over $\{0,1\}$ w/ $\Pr[X=1]=p$, $\Pr[X=0]=1-p$ [indicator random variable] [coin toss] [single sample detecting variant]

[how long till $X=1$?]

def: $X_1, X_2, \ldots, X_n, \ldots$ bernoulli r/v param p, independent. A geometric random var w/ param p is $Y=\min\{i : X_i = 1\}\in\mathbb{N}$ [waiting time for first success] [need ≥ 1 test]

lem:
- $Y \geq 1$
- $\Pr[Y\geq i] = \Pr[X_1 = \cdots = X_{i-1} = 0] \overset{\text{indep}}{=} (1-p)^{i-1}$
- [E?] [pick lemma] $\Pr[Y=i] = \Pr[\quad \text{"} \quad, X_i = 1] = p(1-p)^{i-1}$

Lem: Z rand var over $\mathbb{N}$
- (a) $\mathbb{E}[Z] = \sum_{i=0}^{\infty} i\Pr[Z=i]$
- (b) $\mathbb{E}[Z] = \sum_{i=1}^{\infty} \Pr[Z\geq i]$

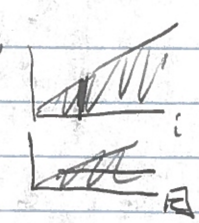
pf: (a): $\mathbb{E}[Z] \overset{\text{[defn]}}{=} \sum_{\omega\in\Omega} Z(\omega)\Pr[\omega]$

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

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

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

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



2022 [illegible]

cor: $\mathbb{E}[\mathrm{Geom}(p)] = 1/p$

pf: $= \sum_{i\geq 1} \underbrace{\Pr[\mathrm{Geom}(p)\geq i]}_{=(1-p)^{i-1}} = 1 + (1-p) + (1-p)^2 + \cdots \overset{\text{[geom series]}}{=} \frac{1}{1-(1-p)} = 1/p$ ∎

[rare things hard to find]

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

[wlog for easier discussion]

def: $a=(a_1,\ldots,a_n)$ distinct integers
$\mathrm{rank}(a_i) = \{\# j : a_j < a_i\} + 1$

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

sketch: $\boxed{\quad a \quad} = \boxed{\quad b \quad}\boxed{a_i}\boxed{\quad c \quad}$ $\quad b_j < a_i < c_k$ any j, k

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

Michael A. Forbes
miforbes@illinois.edu
2024-02-29.2

def: given $a=(a_1,\ldots,a_n)$, $1\le m\le n$, the selection problem is to output a_i s.t. $\mathrm{rank}(a_i)=m$. median problem if $m=\lfloor n/2\rfloor$

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

algo: sort a in $O(n\lg n)$ time [[CS374]] [[merge sort]]
output m-th element of sorted order.

[] [[cannot improve as stated]]

- derivation
fact: [comparison] sorting algo [[def]] require $\Omega(n\lg n)$ time

- dec $O(n)$ vs $O(n)$ dec
fact [CS374]: selection can be done in deterministic $O(n)$ time [[median of medians]]
[pf] not "simple" [[O(1) vs O(lg n) sometimes nontrivial]]
poor constants [[not "practical"]]

[[better?]]

thm: selection in $O(n)$ [expected] time [[simple]] [[good constants]]
[[make formal this]]

splitter
algo: some-select(a,m) = – use [some] rule to pick splitter a_i
– write $a = b, a_i, c$ $\quad b_j < a_i < c_k$ all j,k
$\boxed{\quad a\quad} = \boxed{b}\boxed{a_i}\boxed{c}$, $\mathrm{rank}(a_i)=|b|+1$
– if $m=|b|+1$, return a_i [[fits]]
– if $<$, return some-select(b,m) [[fits]]
– if $>$, $\quad$ " $\quad(c, m-(|b|+1))$ [[fits]]
[[clear]]

[[correctness]]
prop: [any] method to pick splitter is correct

[[complexity]]
vs 2022
prop: suppose can pick splitter in $O(n)$ time deterministically [[warm-up]]
then some-select runs in $O(n^2)$ time [[worse than sorting]]

pf: $T(a)$ [[time for a, max over m]] $\|\le \underbrace{O(n)}_{\text{find splitter}} + \underbrace{O(n)}_{\text{split}}$ [[move a_i to correct spot]] $+ \underbrace{\max\{T(b),T(c)\}}_{\text{recursive call}}$

$= \max_i T(a,i)$

$T(n) \;\|\le O(n) + T(n-1)$ [[solve]] $\qquad |a|=|b|+1+|c|$
$=\max_{|a|\le n} T(a) \quad \le \ldots \qquad \Rightarrow |b|,|c|<|a|=n$
$\le O(n^2)$ [[general strategy]]

[[better?]]
idea: pick [balanced] split
$\hookrightarrow \mathrm{rank}(a_i)\in\left[\frac{n}{4},\frac{3n}{4}\right]$ [[perfect balance]] [[easier to do, still enough]]

$\boxed{a} = \boxed{b}\boxed{a_i}\boxed{\quad c\quad}$ [[unbalanced]]
$\boxed{\quad a\quad} = \boxed{b}\boxed{a_i}\boxed{c}$ [[balanced]]
$\Rightarrow |b|,|c|\le \frac{3n}{4}$ [[better recursion]]

- recursion
[but] is selection problem! [[circular]]

idea [any] a_i w/ suffice
[half] of a_i have

[[w/p 1/2 succeed]]

$\frac14 \quad \frac34$
sort(a)
good splitters

$\rightarrow$ pick a_i [randomly]

Michael A. Forbes
MiForbes@illinois.edu
2024-02-29.3

algo: pick-splitter(a) - while
- $O(1)$ - pick $i \in \{1, \dots, n\}$ randomly
- $O(n)$ - write $a = b, a_i, c$ $\quad b_j < a_i < c_k$ all j,k
- $O(n)$ - if $\underbrace{\text{rank}(a_i)}_{=|b|+1} \in [n/4, 3n/4]$, return a_i $\quad$ clear.

[correctness is clear]

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

pf: L = # of loops

[accounting]

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

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

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

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

□

cor: same-select w/ pick-splitter has $O(n)$ expected runtime

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

[nontrivial!] $\quad$ [random vars]

linearity of expectation [any a,m] ⇒

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

$|b|, |c| \le \frac{3n}{4} + 1$ $\quad$ $\le T(\frac{3n}{4})$ [notation]

$$T(n) \le O(n) + T(3n/4)$$ [solve]

$\le \dots$

$\le O(n)$

$= \max_{\substack{|d| \le \frac{3n}{4} \\ 1 \le \ell \le |d|}} \mathbb{E}[T(d,\ell)]$

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

rmk: "simple" algo, [but] analysis more difficult $\quad$ Mulmuley .

good constants

Q: selection vs sorting?

↳ [one] element in sorted order $\quad$ ↝ [all] elements in sorted order $\quad$ [sort ⇒ select] [converse??]

algo: quicksort(a):
- use [some] rule to pick splitter a_i
- write $a = b, a_i, c$ $\quad b_j < a_i < c_k$ all j,k
- return quicksort(b), a_i, quicksort(c) $\quad$ return .

[analogous] $\quad$ analog .

[correctness]

prop: quicksort always sorts correctly

[complexity] $\quad$ [warm up] $\quad$ vs 2022 .

prop: suppose choosing splitter in $O(n)$ deterministic time

then quicksort runs in $O(n^2)$ time deterministically $\quad$ [any rule]

[do both]

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

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

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

Michael A. Forbes
miforbes@illinois.edu
2024-02-29.4

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

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

$\leq O(n^2)$ ☐

rmk: [is] $\Theta(n^2)$ if $m=1$ always

$\Rightarrow T(n) = \Theta(n) + T(n-1) + T(0) = \cdots = \Theta(n^2)$, with $T(0) = 0$

[[general paradigm]]

idea: [balanced] split

↳ at median via select$(a, n/2)$ [[we solved this]]

cor: quicksort [[randomized]] w/ splitter as [median] takes $O(n \lg n)$ expected time

pf. $T(a) \leq \underbrace{M(a)}_{\text{median}} + \underbrace{O(n)}_{\text{split}} + T(b) + T(c)$

[[linear]]

$\mathbb{E}[T(a)] \leq \mathbb{E}[M(a)] + O(n) + \mathbb{E}[T(b)] + \mathbb{E}[T(c)]$

[[rand]]
$|b| = |c| = n/2$

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

[[max over a]]

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

$\leq \cdots$

$\leq O(n \lg n)$ ☐

• motivation

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

today: rand algo
- geometric rand var
- randomized selection
- randomized quicksort

reading: KT 13.3, 13.5

next lecture: rand algo

logistics: pset 5 out F17