Advanced Algorithms

Lecture #1 1/12/2021

- self evaluation test.

---

2 diff from AoA.

1) Randomized: Pr[algo correct] $\geq 90\%$.

2) Approx: outputs a'

$$a \leq a' \leq c \cdot a$$

approx. correct ans.

1+2: Pr[algo outputs a' $a \leq a' \leq c \cdot a$] $\geq 90\%$

c = approx. ratio 1% → $c = 1.01$

$c = 2$

$c = f$(input size).

---

1) Design algo

2) Analysis a) correctness

b) performance → time.

→ space.

→ communicat[ion] network us[age]

---

1) Hashing: $h: U \to [n] = \{1, 2, \dots n\}$.

solves <u>Dictionary problem</u>

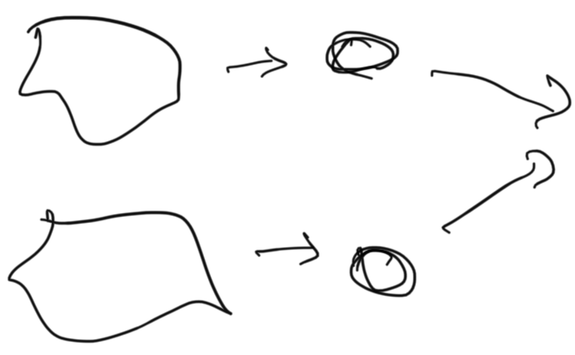
$O(1)$ expected query time

<u>perfect hashing</u>: $O(1)$ w.c. q.t.

2) <u>Sketching / streaming algos.</u>

↓ functional compression

a)



similar or not.

b) traffic

store synopsis of traff[ic]

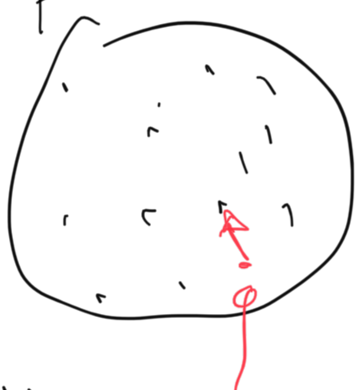
ex: estimate # distinct IPs seen.

related to high dim. geom.

$\mathbb{R}^d$ d "large".

3) Nearest Neighbor Search (NNS).
in high. d. space

P



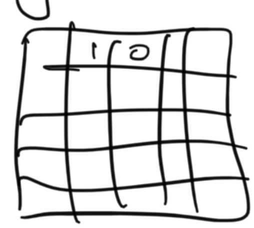
4) Graphs → advanced algos/methods for graphs.

Max-Flow: - poly-time algo.
- scaling algo's.

5) Spectral graph theory.

G

A_G =



spectral clustering algo.

6) Optimization.

$$\min_{\substack{x \in \mathbb{R}^d \\ x \in C}} f(x)$$

Lin. Programming: f, C → linear

Interior Point Method

↳ gradient descent, Newton's method,

Multiplicative Weights Update.
learning from experts.

7) Large-scale models.
- parallel/cluster computing, for systems like MapReduce
- I/O external model
  → takes into consid cache.

8) etc.

Course Expectations / Del.:

Grading:
1) scribing 10%
2) 5 hw's 55%

5 "free" late days.

10% off / day. → 5 days la

3) project. 35% team

- report 1 } light.
- — " — 2 }
- final. 25%

Types:
1) reading
2) implementation.
3) research-level.

Prereq: - strong math backg.

- basic prob., linear algebl

- CS backgr.
  - $O(\cdot)$, $\Omega(\cdot)$.

- sorting, bin. search, coun. comp.

Problem: counting up to n.

space: how many bits will be nec. to count to n.

$\lceil \log_2 n \rceil$ bits to repr. n.

Goal: to do better.

Thm: can't do better if algo det. or is exact.

Morris's Algo: get $O(\lg \lg n)$ bi
to count up to n, up to
constant-factor approx.
$\rightarrow c = O(1)$

M.a.: - counter $X = 0$.

- @ button: $X := \begin{cases} X+1, & \text{with pr} \\ & 2^{-X} \\ X, & \text{otherwise} \end{cases}$

- @ end, to output approx to
  # presses of button:

$\hat{n} = 2^X - 1$

$\hat{}$ the estimate.

$$n = \underbrace{\boxed{1\,|\,1\,|\,0\,|\,1\,|\,1\,1\ldots|\,.}}_{\lceil \lg_2 n \rceil}$$

$\lceil \lg_2 n \rceil \leftarrow$ X approximate

$\Rightarrow$ to repr. X need a $O(\lg \lg$

bits

<u>Analysis:</u> a) Correctness:

$$\Pr[\hat{n} \approx n] \geq 90\%.$$

<u>Probability:</u>

$X \rightarrow$ random variable.

<u>Def:</u> $\mathbb{E}[X] = \sum_a a \cdot \Pr[X = a]. \leftarrow$

$$\int_a$$

<u>Def:</u> $X = X_1 + X_2$.

$$\mathbb{E}[X] = \mathbb{E}[X_1] + \mathbb{E}[X_2].$$

<u>Concentration bounds:</u>

Lemma: (Markov's bound)

Lemma: [Markov's bound]

$X \geq 0$: $\forall \lambda \geq 0$:

$$\Pr[X > \lambda] \leq \frac{\mathbb{E}[X]}{\lambda}.$$

$$\Pr[X \geq c \cdot \mathbb{E}[X]] \leq 1/c.$$

Def: Variance: $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2]$

Lemma [Chebyshev's bound]: $\forall \lambda > 0$.

$$\Pr[|X - \mathbb{E}[X]| > \lambda] \leq \frac{\mathrm{Var}[X]}{\lambda^2}.$$

↳ close or = to actual ans.