03/12 350 Lec1

1. Intro

a. common techniques

(Recursion)
- Divide & Conquer
- Greedy
- dynamic programing

+ exposure to complexity theory.

What I'll do:

→ Modern algorithms
- Randomized algorithms
- Approximation algorithms
- online/streaming algorithms

1. Randomized alg's.

a. Warm-up:

Computation needs resources:

- time
- space ✓

computational ✓

$\boxed{\text{Randomness is a resource}}$

^^ often extremly simple algorithms.

^^ outperm deterministic algorithms on time/space

☹ fail sometimes.

prob. of failure $\longrightarrow 0$ $\frac{1}{2^{100}}$

vs. prob. of hardware failure ?

vs. prob. of computer hit by a meteor $\frac{1}{2^{60}}$

b. Examples.

- Primality testing (素数判定)

  → direct alg.

  Given: integer $N > 0$ len(N) $\overset{\text{binary}}{=} \log N = n$

  Goal: decide N prime ?

  - direct algorithm: $2, 3, 4, \dots, \sqrt{N} \overset{?}{\mid} N$

  $O(N) = O(2^{n/2})$

→ 70's: efficient rand alg. poly(n)

[Miller-Rabin / Solovay-Strassen].

→ [AKS'04] poly-time deterministic alg

★ higher poly, complicated, Real. apps.

- Polynomial identity Testing (PIT)

→ poly-time det. alg. unknown

^^ $\exists$ very efficient rand. alg.

- Matrix - Product checking.

Given: 3 $n \times n$ matrices A, B, C

Goal: decide if $AB = C$.

→ Det. alg: $A \cdot B$ matrix mult.

Compare w/ C.

M.M alg's: naive $O(n^3)$

Strassen $O(n^{2.81})$

(Div.&Con)

$O(n^{2.37188})$ 2022

$\omega(n^2)$

→ Rand. alg: $O(n^2)$ time

$\boxed{\begin{array}{l} \mathcal{A}: \text{ on input } A, B, C \\ \quad \bullet\ \underline{\text{sample}}\ r \in \{0,1\}^n \text{ unif. at rand.} \\ \quad \bullet\ \text{Output: } A\left(\cdot B \begin{pmatrix} | \\ r \\ | \end{pmatrix}\right) \stackrel{?}{=} C\begin{pmatrix} | \\ r \\ | \end{pmatrix} \end{array}}$

<u>Time</u>:

$$B \cdot \begin{pmatrix} | \\ r \\ | \end{pmatrix} \to r' : \quad O(n^2)$$

$$A \cdot r' \to r'' : \quad O(n^2)$$

$$C \cdot r \to r''' : \quad O(n^2)$$

$$r'' \stackrel{?}{\longleftrightarrow} r''' \qquad O(n)$$

$$\to \quad O(n^2)$$

error : $r = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

if $AB = C$: always correct

if $AB \neq C$: $\underline{\Pr_r[AB \cdot r = C \cdot r] \leq \frac{1}{2}}$

2. Probability Review.

a. Basics.

- Sample space Ω : set of all possible out comes.

1 正 反 0

  - A single coin toss : $\Omega = \{H, T\}$
  - Two coin tosses : $\Omega = \{HH, HT, TH, TT\}$
  - Roll a 6-sided die : $\Omega = \{1, 2, \dots, 6\}$

- DEF: an event E is any subset of Ω
  - getting at least one Head in 2 tosses

    $\Omega = \{HH, HT, TH, TT\}$.

    $E = \{HH, HT, TH\}$.

- DEF: E & F mutually exclusive if $E \cap F = \phi$.
  - E : roll an even number $\{2, 4, 6\}$
  - F : roll an odd number $\{1, 3, 5\}$

- Consider a function

$$P : \Omega \to [0,1]$$
$$\omega \mapsto P(\omega) \quad Pr(\omega)$$

- Ex: $\Omega = \{H, T\}$.

  what do you want P to be?

  - <u>fair coin</u>: $P(H) = P(T) = \frac{1}{2}$
  - <u>Biased coin</u>: $P(H) = 0.85$
    $P(T) = 0.15$

  <u>OBS</u>: $P(H) + P(T) = 1$

- Ex: 2 coin tosses $\Omega = \{HH, HT, TH, TT\}$.

  → $P(\quad) = P(\quad) = P(\quad) = P(\quad) = \frac{1}{4}$

  → Prob. of some outcome, i.e.

  $E = \{HH, TT\}$ NOT an elem. of Ω

  $$P(E) = P(HH) + P(TT) = \sum_{\omega \in E} Pr(\omega) = \frac{1}{2}.$$

- <u>DEF</u>: A <u>probability space</u> is a pair

  (Ω, P)

- Ω : sample space
- P : prob. function

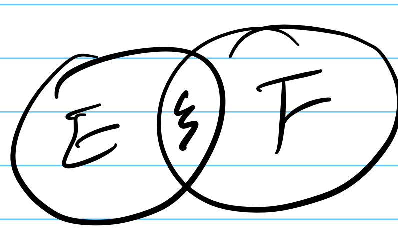
$$\Omega \to [0,1]$$

- $P(w) \geq 0 \quad \forall w \in \Omega$
- $\sum_{w \in \Omega} P(w) = 1$
- if events E & F mutually exclusive.

$$P(E \cup F) = P(E) + P(F)$$

Axioms of Probability.

- Corollaries.
  - $P(\bar{E}) = 1 - P(E)$ (complement) 补
  - if $E \subseteq F$: $P(E) \leq P(F)$
  - $P(E \cup F) = P(E) + P(F) - P(E \cap F)$



- Equally likely outcomes

$$P(\omega) = \frac{1}{|\Omega|} \quad \forall \omega \in \Omega$$

$$\Rightarrow \quad P(E) = \frac{|E|}{|\Omega|}$$

Ex: Toss a coin 100 times.

what's prob. 50 H's ?

A. $1/2$. B. $\frac{1}{2^{50}}$ C. $\binom{100}{50} / 2^{100}$

03/13 350 Lec 2

1. Probability cont'd.

a. More exercises.

- Roll 2 dice (6-sided)

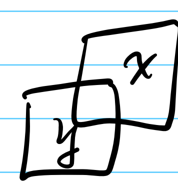
  what's probability both being <u>even</u>?

  - $\Omega = \{1, \cdots, 6\} \times \{1, \cdots, 6\}$
  - $P(\omega) = \frac{1}{|\Omega|} = \frac{1}{36}$, $\forall \omega \in \Omega$
  - $E := \{2,4,6\} \times \{2,4,6\}$
  - $P(E) = \frac{|E|}{|\Omega|} = \frac{9}{36} = \frac{1}{4}$.

- Shuffle a deck of cards (52)

  - Any arrangement equally likely

  ? Probability <u>top two cards</u> have same <u>rank</u>



  - $\Omega = \{(x, y) : x, y \text{ top 2 cards}\}$
  - $P(\omega) = \frac{1}{|\Omega|} = \frac{1}{52 \times 51}$ $\forall \omega \in \Omega$

- $E := \{(x,y) : x, y \text{ have same } \underline{\text{rank}}\}$

$$|E| = \binom{13}{1} \cdot \underbrace{4 \cdot 3}_{P(4,2)}$$

- $P(E) = \dfrac{|E|}{|\Omega|} = \ldots$

b. Conditional probability.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$(P(B) \neq 0)$

$$P(A \cap B) = P(A|B) \cdot P(B)$$

? $P(A|B) = P(B|A)$

c. independence.

· <u>DEF</u>: A & B <u>independent</u>:

$$P(A \cap B) = P(A) \cdot P(B).$$

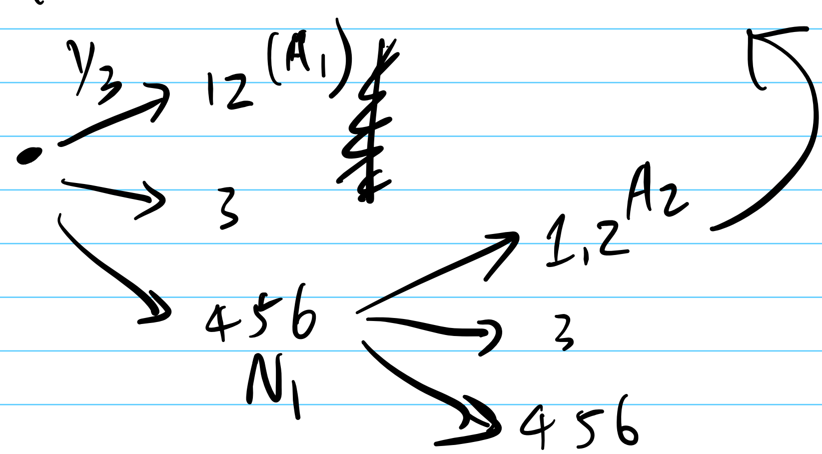
$$\Longleftrightarrow P(A) = P(A|B)$$

• Ex: Roll a 6-sided die.

→ 1 & 2 → Alice wins
  3 → Bob wins
stop

otherwise, another round.

- $P(\text{Alice wins } 1^{st} \text{ round}) = 1/3$
- $P(\text{Alice} \cdots 2^{nd} \text{ round}) = \frac{1}{6}$



$$P(A_2) = P(N_1 \cap A_2)$$
$$= P(N_1) \times P(A_2 | N_1)$$
$$= \frac{1}{2} \times \frac{1}{3}$$

- If they keep playing

$P(\text{Alice wins at some point}) \stackrel{?}{=} 2/3$

2. Analyze MPC

Given: A, B, C $n \times n$

Goal: $AB = C$.

> $\mathcal{A}$:
>
> Choose random $r \leftarrow \{0,1\}^n$
>
> check $A \cdot (B(r)) \stackrel{?}{=} C(r)$

If $AB = C$, always correct

if $AB \neq C$, incorrect when $ABr = Cr$

Claim: $\Pr_{r \leftarrow \{0,1\}^n}(ABr = Cr) \leq \frac{1}{2}$

Pf: Let $\underset{\neq 0}{D} = AB - C$ $(AB \neq C)$

$z = D \cdot r$ $\qquad$ $d_{ij} \neq 0$ for some i, j.

$$\begin{array}{cc} & D \quad\quad j \\ i & \begin{pmatrix} \cdots\cdots d_{ij} \cdots \end{pmatrix} \end{array} \begin{pmatrix} | \\ r \\ | \end{pmatrix} = \begin{pmatrix} | \\ z_i \\ | \end{pmatrix} \rightarrow$$

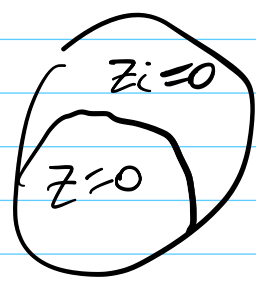
$$z_i = \boxed{\sum_{k=1}^{n} d_{ik} r_k = 0}$$

$$\Leftrightarrow \underset{\neq 0}{d_{ij}}\, r_j = \sum_{k \neq j} d_{ik} r_k$$

$$\Leftrightarrow \boxed{r_j = \frac{1}{d_{ij}} \sum_{k \neq j} \underset{x}{d_{ik} r_k}}$$

$$P[r_j = \underset{\text{indep of } r_j}{x}]_{z_i = 0} = \frac{1}{2}.$$

<u>To fail</u>: $\forall i\ z_i = 0$.



$$P(z=0) \leq P(z_i = 0) = \frac{1}{2}.$$

3. Finger printing

<u>Given</u>: x & y.

<u>Goal</u>: decide if $x = y$ w/ few bits exchange.

$\rightarrow$ Det: msg space $O(n)$

→ **Randomized**: $O(\log n)$ bits

pick $p \le c \cdot n \cdot \log n$	
send $x' := x \bmod p,\ p$ $\longrightarrow$	output $x' \overset{?}{=} y \bmod p$

Analysis:

if $x = y$: $x' = y'$ always holds

$x \neq y$: fail $\quad \underbrace{x \bmod p = y \bmod p}$

$\Leftrightarrow \boxed{p \mid x - y \le 2^n}$

FACT: Prime number theorem.

$$\left| \{ p : p \le a \ \& \ \text{prime} \} \right| \sim a / \log a$$

$\Rightarrow$ # primes $\le c \cdot n \log n$

$$\sim \frac{c n \log n}{\log(c n \log n)} \sim c n$$

$$\Rightarrow \underset{p}{P}(p \mid x - y) \le \frac{n}{c n} = \frac{1}{c}.$$

$$\le \frac{1}{4} \text{ if } c > 4 \quad \#$$

4. Coupon Colleztor problem.

盲盒: Ne zha.

→ 20 items

→ every time pick one at random.

? how many times to zollezt all

A. 20, B. 40 C. 20 . log20

$\approx 20 \cdot 4.3 = 86$

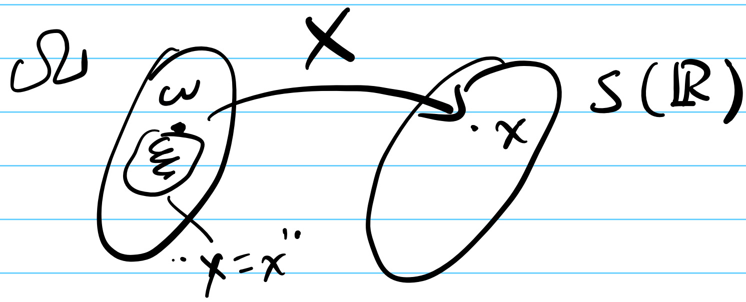
03/19 350 Lec 3

1. Random variables

a. Basics

- coin tosses: # of heads.
  - random return HW assignments
    # students get their own assignment.

- **DEF**: $X : \Omega \to S(\mathbb{R})$



  - $X = x$: event $E := \{\omega : X(\omega) = x\}$

- **Indep**. R.V's. X, Y R.V's.
  X & Y are called **indep**. iff.
  for all possible x & y.
  events $X = x$ & $Y = y$ **indep**.

- Expectation (期望): weighted average.

$$\mathbb{E}[X] = \sum_{x \in S} P(X = x) \cdot x$$

★: **Linearity of expectation (LoE)**

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

- **Ex.** Bet game
  - You choose a card from a deck
  - I pay you ¥ $\begin{cases} 5 & ♡ \\ 0 & \text{o.w.} \end{cases}$
  - X: your earning.

$\Omega = \{ ♡, \overline{♡} \}$.

$X : \Omega \to \mathbb{R}$

$\omega \longmapsto X(\omega) \in \{0, 5\}$

Ω	X	$P(X=x)$
♡	5	1/4
$\overline{♡}$	0	3/4

$$\mathbb{E}[X] = \sum_{x \in S} P(X=x) \cdot x$$

$$= P(X=5) \cdot 5 + \underbrace{P(X=0) \cdot 0}_{=0}$$

$$= 1/4 \cdot 5 = 5/4 \quad \#$$

- Ex: same setup.
  - play it 100 rounds
  - Y: total earning.

$\mathbb{E}[Y] = ?$

$\rightarrow$ (LoE)

Let X_i = earning in ith round.

$i = 1, \ldots, 100$

OBS: $Y = X_1 + X_2 + \cdots + X_{100}$

$\cdot\ \forall i: \quad \mathbb{E}[X_i] = 5/4$

$$\Rightarrow \mathbb{E}[Y] = \mathbb{E}[X_1 + \cdots + X_{100}]$$

$$\xrightarrow{\text{(LoE)}} = \mathbb{E}[X_1] + \mathbb{E}[X_2] + \cdots + \mathbb{E}[X_{100}]$$

$$= 100 \times 5/4 = 125$$

b. useful R.V's.

- Bernoulli R.V. $\longleftrightarrow$ biased coin toss

$$X = \begin{cases} 1 & p \\ 0 & 1-p \end{cases}$$

$$\mathbb{E}[X] = p \cdot 1 + \underline{(1-p)0} = p$$

- Binomial R.V: # of heads n coin tosses w/ bias p

$X \sim Bin(n, p)$

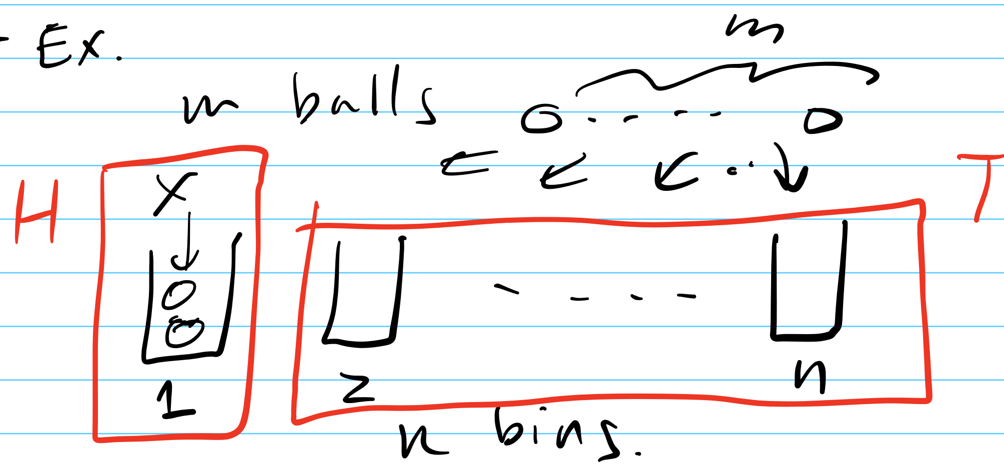
($\sim$: has prob. distribution)

$$P[X=k] = \binom{n}{k} \cdot p^k (1-p)^{n-k} \quad \forall k = 0, \dots, n$$

$$E[X] = \sum_{k=0}^{n} P(X=k) \cdot k$$

$$\boxed{= n \cdot p}$$

- Ex.



① $X := \#$ balls in bin 1

$X \sim Bin(m, \frac{1}{n}) \quad p = \frac{1}{n}$

$E[X] = m \cdot \frac{1}{n} = \frac{m}{n}$

② $Y := \#$ empty bins.

$E[Y] = ?$

Ex.

- Ex: (Return HW).

$X := $ # student get their own assign.

students $= n$.

$i = 1, \cdots, n$

$$X_i := \begin{cases} 1 & \text{if student } i \text{ get correct HW.} \\ 0 & \text{o.w.} \end{cases}$$

O.BS:

- $X = X_1 + \cdots + X_n$
- $\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i]$

$\forall i, \quad \mathbb{E}[X_i] = P(X=1)\cdot 1 + P(X=0)\cdot 0$

$= \frac{1}{n}$

$= n \times \frac{1}{n} = 1$

- Geometric R.V. X

$$\overbrace{\underset{T}{O} \ \underset{T}{O} \ \cdots \ \underset{T}{O} \ \underset{H}{O}}^{X}$$

$H : p$

$T : 1-p$

X: # ~~tosses~~ till first see "H"

$X \sim \text{Geom}(p)$

Claim: $P[X=n] = (1-p)^{n-1} \cdot p$

$\forall\, n = 1, 2, \cdots$

Claim: $\mathbb{E}[X] = \sum_{n=1}^{\infty} P[X=n] \cdot n$

$\boxed{= 1/p}$

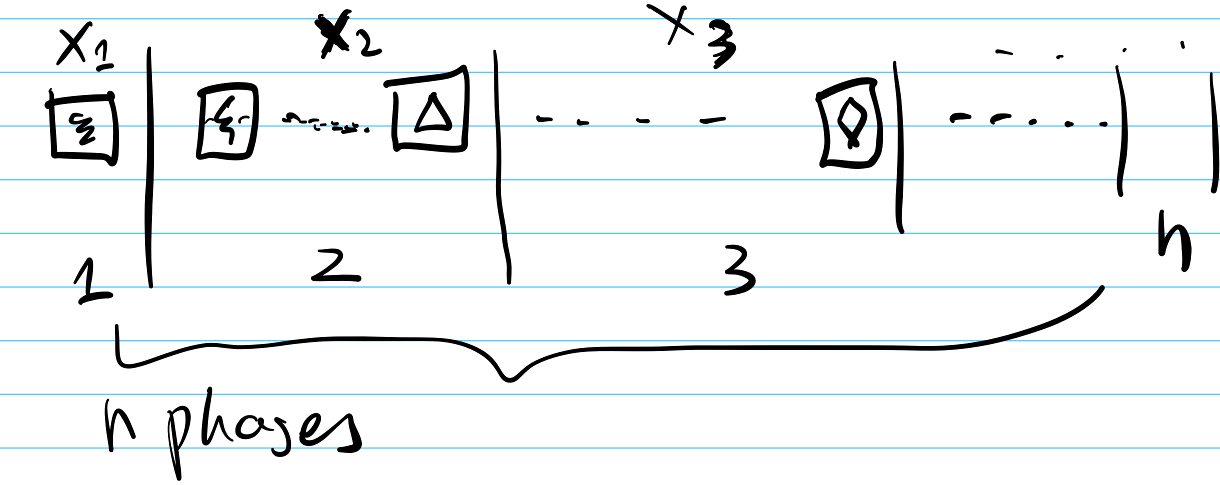
#

2. Coupon Collector problem:

a. n coupons

- every time get one coupon uniformly at random
- $X :=$ # of purchases / boxes to collect at least one copy of each coupon

?. $\mathbb{E}[X]$



$X_i :=$ already had $(i-1)$ coupons

$\#$ boxes to get a new coupon.

$$X = \sum X_i = X_1 + \cdots + X_n.$$

$X_i \sim \text{Geom}(p_i)$ (Geometric R.V)

$i=2,\ p_2 = \frac{n-1}{n}$

$i=3\ \ p_3 = \frac{n-2}{n}$

$$p_i = \frac{n-(i-1)}{n}$$

$$\mathbb{E}[X_i] = 1/p_i = \frac{n}{n-(i-1)}$$

$$\Rightarrow \mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i]$$

$$= \sum_{i=1}^{n} \frac{n}{n-(i-1)}$$

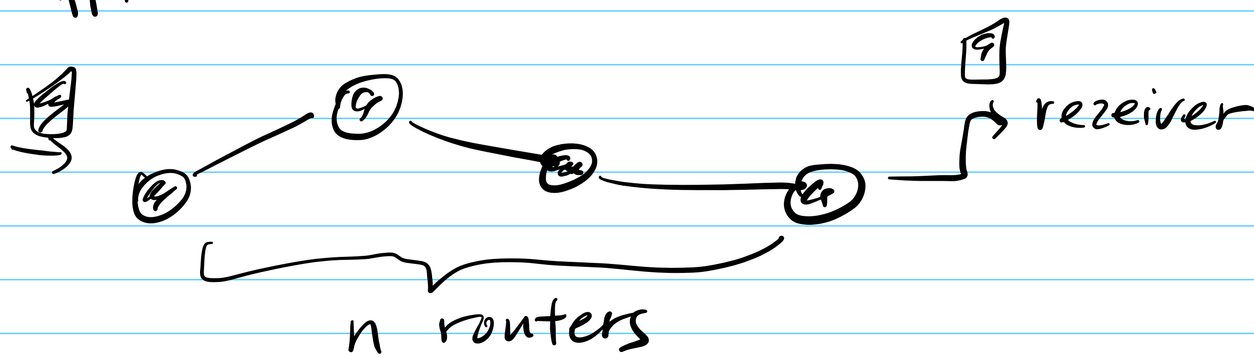
$$= \frac{n}{n} + \frac{n}{n-1} + \frac{n}{n-2} + \dots + \frac{n}{1}$$

$$= n \cdot \underbrace{\sum_{k=1}^{n} \frac{1}{k}}_{= H(n)}$$

FACT: $\ln n \leq H(n) \leq \ln n + 1$

$$E[X] \approx n \cdot \ln n$$

b. Application



Goal: Receiver want to know all n routers.

- package has space for one name & counter

Idea: $\boxed{\text{Sample a uniform router}}$

??? distributed setting. ☆

03/20 350 Lec4

1. Intro: Vertex cover

a. Basics.

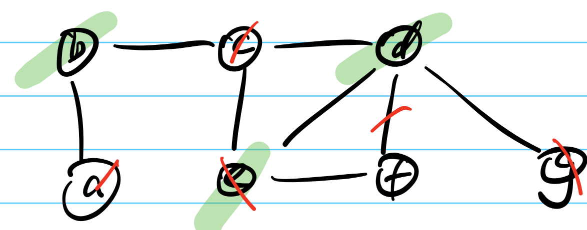
nodes/vertex

Graph: $G = (V, E)$

edges

· DEF: A vertex cover $S \subseteq V$ is subset of V s.t. touches all edges

Ex:



· V itself ✓
· (a, c, e, g, d) ✓
· (b, e, d) optimal VC

· Vertex cover

Given: $G = (V, E)$

Goal: find vertex cover S of minimum size

what's known? $|V| = n$

→ Brute force: all subsets of V $O(2^n)$

→ NP-hard: unknown (unlikely) to admit a poly-time alg's.

b. Approx. alg's for VC.

☆: Greedy strategies:
choose one that appears to be beneficial (up to some mesure) at the moment.

- 1st attempt:
  pick the vertex that touches most edges

App-VC1: on input $G=(V,E)$
for $v \in V$ (in descending order of degrees)
- add v ~~to~~ S (VC candidate)
- delete v & neighbors from G

- Analysis:
  - correctness: S will be VC: all edges are touched.
  - optimality?
    $\exists G$, s.t. $|S| = \Omega(\log n \cdot OPT)$

• 2^{nd} attempt.

App-VC 2: on input $G=(V,E)$

while some $\{u,v\}$ is uncovered

add both u, v to T. (VC candidate)

Ouput T

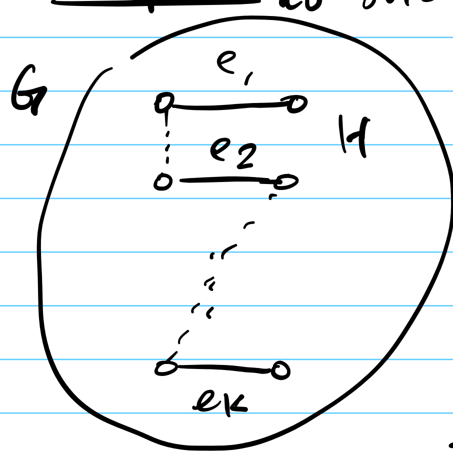
Claim 1: T is VC : by termination condition.

Claim 2: $|T| \le 2 \cdot OPT$ (OPT is size of min. VC)

Pf: let $\{e_1, \cdots, e_k\}$ be the edges chosen during the alg.

$|T| = 2k$.

Suffice to show: $OPT \ge k$



H is subgraph of G. (other edges may exist in G)

At least pick one node from each of k edges in opt VC.

$\Rightarrow OPT \ge k$

$\Rightarrow |T| \le 2 \cdot OPT$ $\blacksquare$

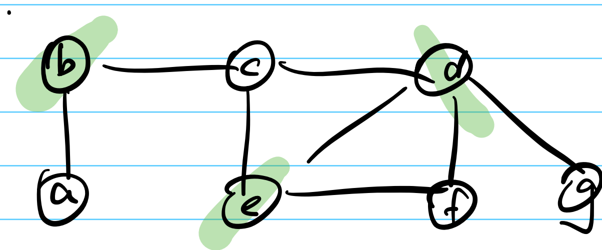
Approximation factor

App-VC2 find 2-Approx. VC
(of size at most 2-OPT)

Ex: Find a graph saturates the bound $|T| = 2\cdot OPT$

This shows 2. approx-factor is tight.

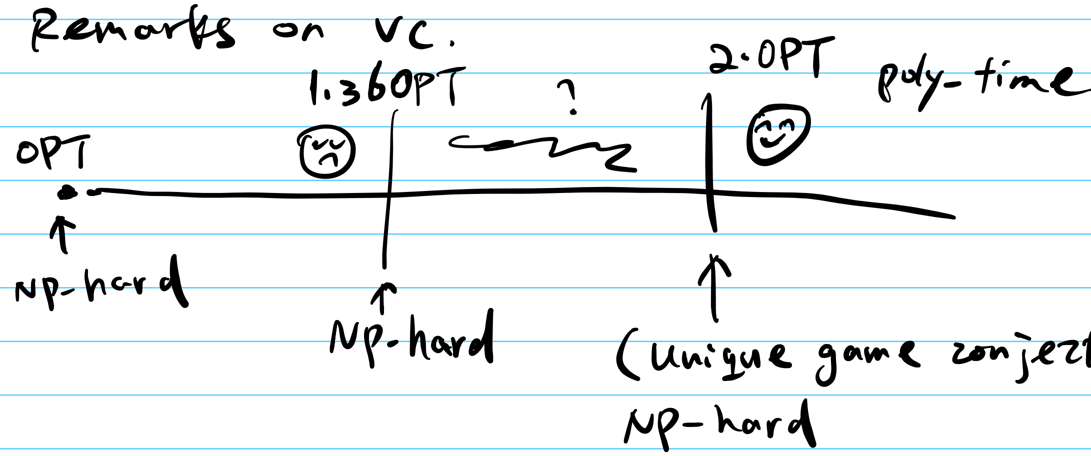
G.



(b, e, d)
OPT = 3

Run App-VC2 on G:

c. Remarks on VC.



· A more principled approach:

Integer Linear Programming ILP.

2. Linear Programming.

a. Basics.

Ex 1.

max: $x_1 + 5x_2$ $(x_1, x_2 \in \mathbb{R})$

(feasible) ✓

LP instance

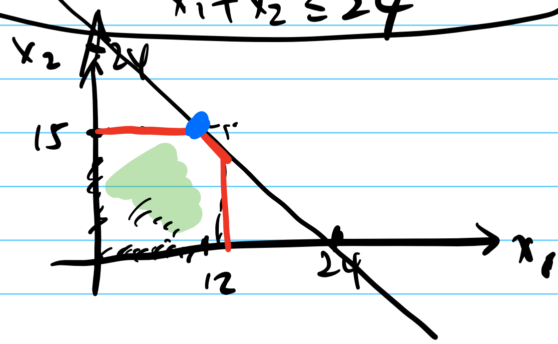
subject to:

$0 \le x_1 \le 12$

$0 \le x_2 \le 15$

$x_1 + x_2 \le 24$

→ linear constraints



$x_1 = 24 - 15 = 9$

$x_2 = 15$

$x_1 + 5x_2 = 9 + 15 \times 5 = 84$

OBS:

$x_1 + 5x_2$

$= \underbrace{(x_1 + x_2)} + 4\underbrace{x_2}$

$\le 24 + 4 \times 15 = 84$

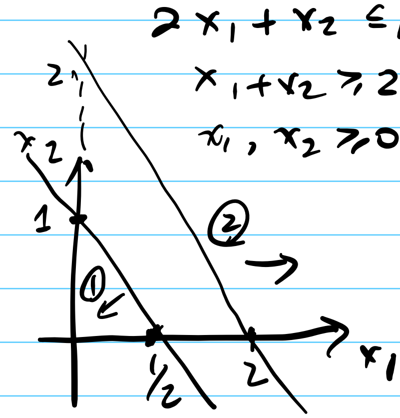
Ex 2. max $x_1 - x_2$ (infeasible)

subject to

$2x_1 + x_2 \le 1$

$x_1 + x_2 \ge 2$

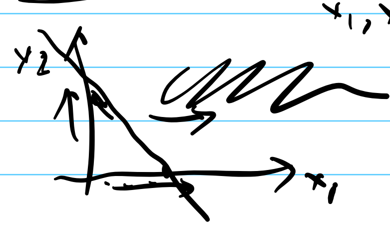
$x_1, x_2 \ge 0$



Ex 3: max $2x_1 + x_2$ → (unbounded)

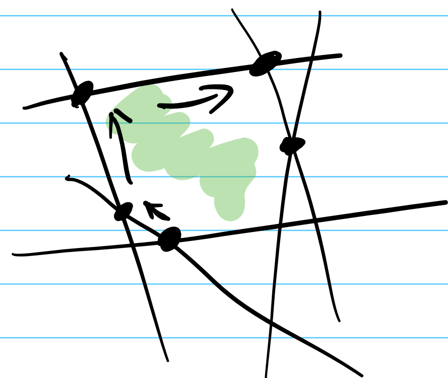
subj. to. $x_1 + x_2 \ge 1$

$x_1, x_2 \ge 0$



b. LP algorithms (for feasible instances)

- Simplex alg. (George Dantzig '1947)



linear constraints define a polygon as feasible region.

Gist: "Hill-climbing" move to a neighbor if obj value increases↗

Details to fill in:

→ how to find initial feasible vertex?

→ which neighbor to move to?

→ Running time?

☹ worst-case exponential time.

☺ super fast real world

→ correctness?

Convex polyhedron + linear obj

⇒ local max = global max

* poly-time LP Alg's

→ Ellipsoid Alg [Khachiyan 1979]
(NOT competitive in practice)

* Interior point alg. [Karmakar 1984]
Naredra

N.B. Commercial solvers
solve LP. w/ millions of variables
& contraints.

c. Formal description

standard form

m : # of constraints $i = 1, \cdots, m$.

n : # of variables $j = 1, \cdots, n$

LP Input: real numbers

c_j, a_{ij}, b_i $i = 1 \cdots m$, $j = 1 \cdots n$

Output: real numbers x_j

max: $\sum_{j=1}^{n} c_j \cdot x_j$ (obj. function)

subject to

$\sum_{j=1}^{n} a_{ij} x_j \le b_i$ $i = 1, \cdots, m$

matrix form $x_j \ge 0$ $j = 1, \cdots n$.

03/26 350 Lec 5

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

objective: $\sum_{j=1}^{n} c_j \cdot x_j$

$c^T \cdot x \quad \langle c, x \rangle$

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}_{m \times n} \longleftarrow$$ A row vector defines one constraint.

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \qquad \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i = 1, \dots, m$$

$$\Updownarrow$$

$$Ax \le b$$

$$x \ge 0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}_{n \times 1}$$

matrix-form LP	
max	$c^T \cdot x$
subj. to:	$Ax \le b$
	$x \ge 0$

$\Downarrow$

a. Integral LP: variables must have
(ILP) integer value.

- Formulating Vertex Cover as ILP
  For each $i \in V$, introduce $x_i \in \{0, 1\}$
  // indicate whether node i is included

ILP for Vertex Cover Π

$$\boxed{\begin{array}{lll} \underline{\min}: & \sum_{i=1}^{n} x_i & (\text{size of VC}) \\ \underline{\text{subj. to}}: & x_i + x_j \geq 1 & \text{for } (i,j) \in E \\ & x_i \in \{0,1\} & \forall i \in V. \end{array}}$$

ILP is NP-hard. poly-time alg. unlikely!

b. VC ILP.

: putting aside integral constraint.

$$\boxed{\begin{array}{lll} \underline{\min.} & \sum_{i=1}^{n} x_i & \\ \underline{\text{subj to}}: & x_i + x_j \geq 1 & \forall (i,j) \in E \\ & 0 \leq x_i \leq 1 & \forall i \in V. \end{array}}$$

LP Σ

Let x^* be an optimal soln. for Σ.

& optimal value $OPT = \sum_{i=1}^{n} x_i^*$

how to derive an integral soln ?

$\boxed{\text{Rounding}}$ (Threshold)

$$\hat{x}_i := \lfloor x_i^* \rceil = \begin{cases} 1 & x_i^* \geq \frac{1}{2} \\ 0 & \text{o.w.} \end{cases}$$

① $\boxed{\{\hat{x}_i\}}$ is a feasible soln ?

$\forall (i,j) \in E,\quad x_i^* + x_j^* \geq 1$

$\Rightarrow x_i^* \geq \frac{1}{2}$ or $x_j^* \geq \frac{1}{2}$ or both.

$\Rightarrow$ after rounding, (i,j) is covered.

② $\sum_{i=1}^{n} \hat{x}_i \leq \sum_{i=1}^{n} 2 \cdot x_i^*$

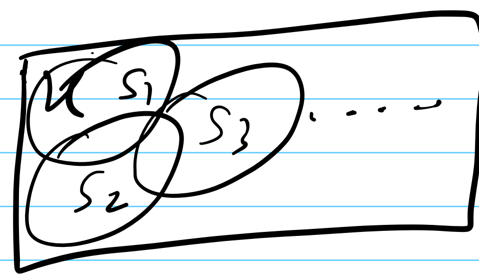
$= 2 \cdot OPT \leq 2 \cdot OPT_{int}$

<u>OBS</u>: $\underset{(LP)}{OPT} \leq \underset{(ILP)}{OPT_{int}}$

C. Set Cover

- <u>Input</u>: $\mathcal{U}$ (universe)

  $S_1, \ldots, S_m \subseteq \mathcal{U}$



<u>Goal</u>: Find $I \subseteq \{1, \ldots, m\}$, as small as possible

s.t. $\bigcup_{i \in I} S_i = \mathcal{U}$.

- ILP Π for set cover.

  for each set S_i introduce $x_i \in \{0,1\}$

  // whether to choose S_i

<u>min</u>: $\sum_{i=1}^{m} x_i$ (# of subsets chosen)

<u>subj. to</u>: $\forall u \in \mathcal{U}$: $\sum_{i:\, u \in S_i} x_i \geq 1$

" at least one of the subsets containing u is chosen.

d. Randomized Rounding

<u>Idea</u>: Suppose x^* is opt. for LP

$0 \le x_i^* \le 1$ (fractional).

$\downarrow$

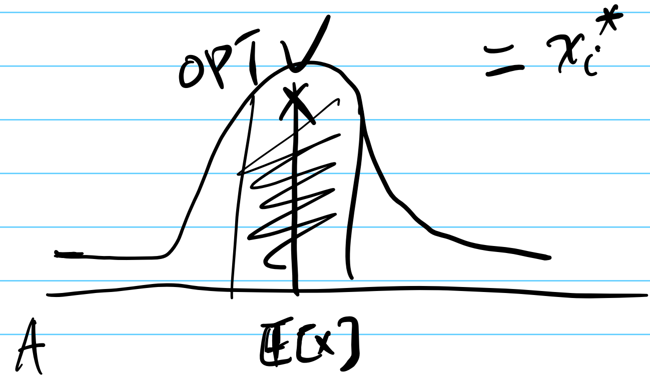
$$\hat{x}_i := \begin{cases} 1 & \text{w.p. } x_i^* \\ 0 & \text{w.p. } (1-x_i^*) \end{cases}$$

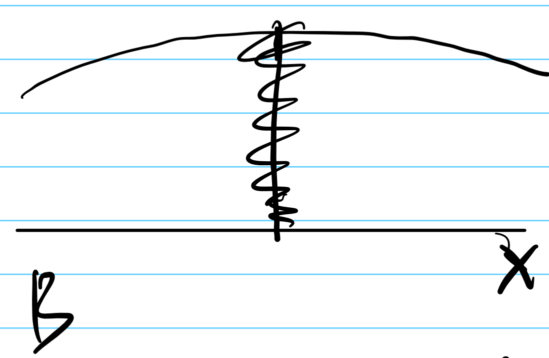
$$\mathbb{E}\left[\sum_{i=1}^{n} \hat{x}_i\right] = \sum_{i=1}^{n} \underline{\mathbb{E}[\hat{x}_i]} = \sum_{i=1}^{n} x_i^*$$

<u>OBS</u>: $\hat{x}_i$ is Bernoulli R.V. $\forall i$ w.p. x_i^*

$$\mathbb{E}[\hat{x}_i] = \underline{1 \cdot P(\hat{x}_i = 1)} + \underbrace{0 \cdot P(\hat{x}_i = 0)}_{=0}$$

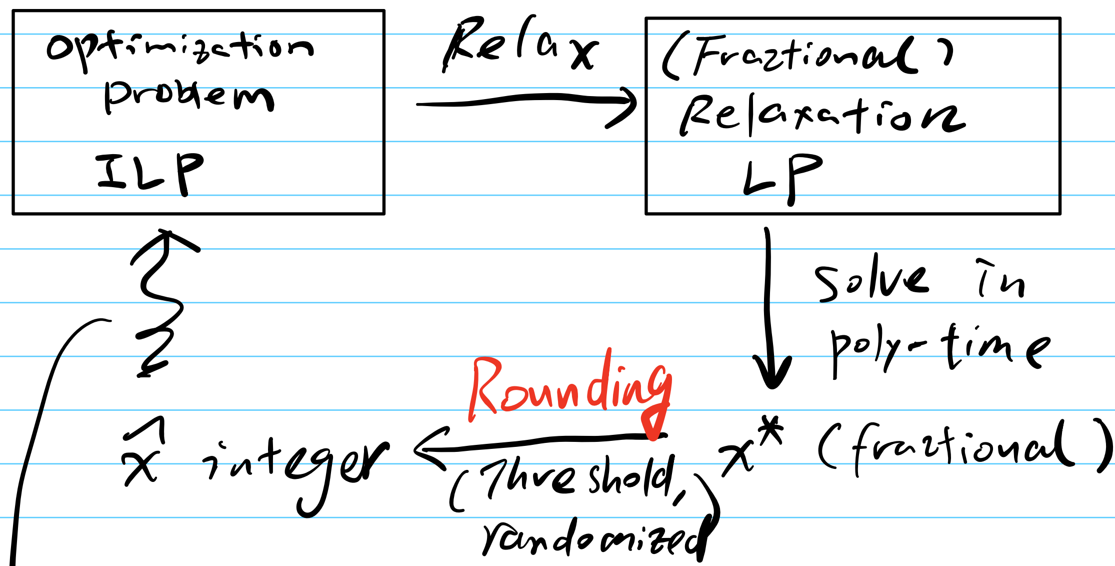
$$= x_i^*$$





03/27 350 Lec26

1. ILP → Approximation algorithms.



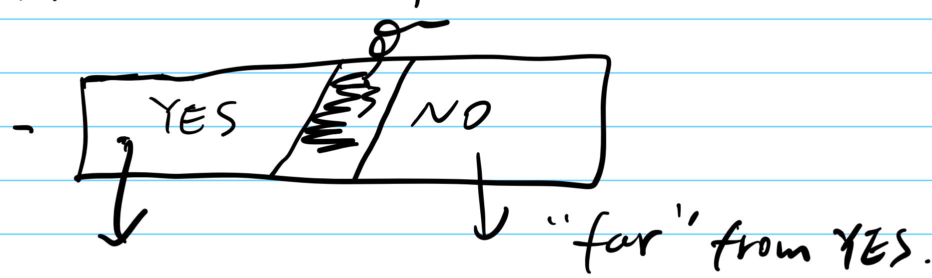
① A valid/feasible sol'n to ILP?

② Any good? (Approximation ratio)

2. Algorithms in new settings: A few examples.

a. Property testing (another kind of approximation)

in P.T.: Decision Problem YES/NO answers

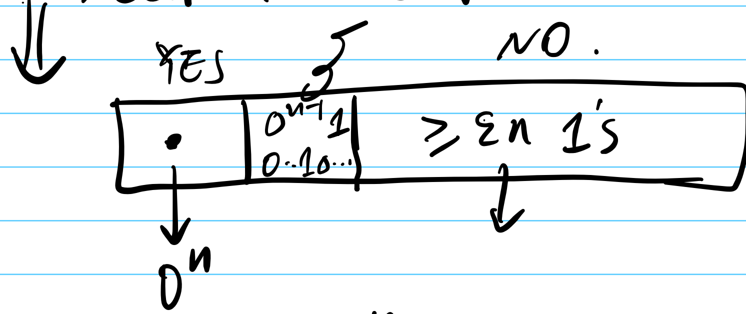


· Toy example: zero-testing

Input: $w \in \{0,1\}^n$

Goal: Decide $w \stackrel{?}{=} 0^n$?

Need to read entire input.



P.T. Is $w=0^n$ or $\geq \varepsilon n$ 1's ("far" from 0^n)?

> Test(n, ε)
> 1. Sample $s\ (=2/\varepsilon)$ positions unif. random
> 2. If "1" found output NO
>   o.w. ---- YES

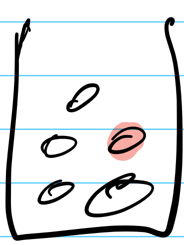
· Analysis:

- if $w=0^n$: always answer YES.
- if w ε-far:

$$P[\text{error}] = P[\text{output YES} \mid |w| \geq \varepsilon n]$$

$$= P[\text{no 1's in } s \text{ random positions}]$$

$$\leq \underbrace{(1-\varepsilon)\cdot(1-\varepsilon)\cdots(1-\varepsilon)}_{s}$$



n balls

at least εn red balls.

$$\underset{(1-x)\leq e^{-x}}{\leq} e^{-\varepsilon \cdot s} = e^{-\varepsilon\cdot\boxed{\frac{2}{\varepsilon}}} = e^{-2} < \frac{1}{3}$$

∎

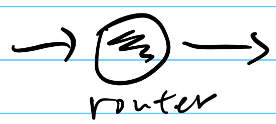
Witness Lemma

If a test catches a witness w.p. $\geq p$ then $s = \frac{2}{p}$ iterations of test catches a witness w.p. $\geq 2/3$

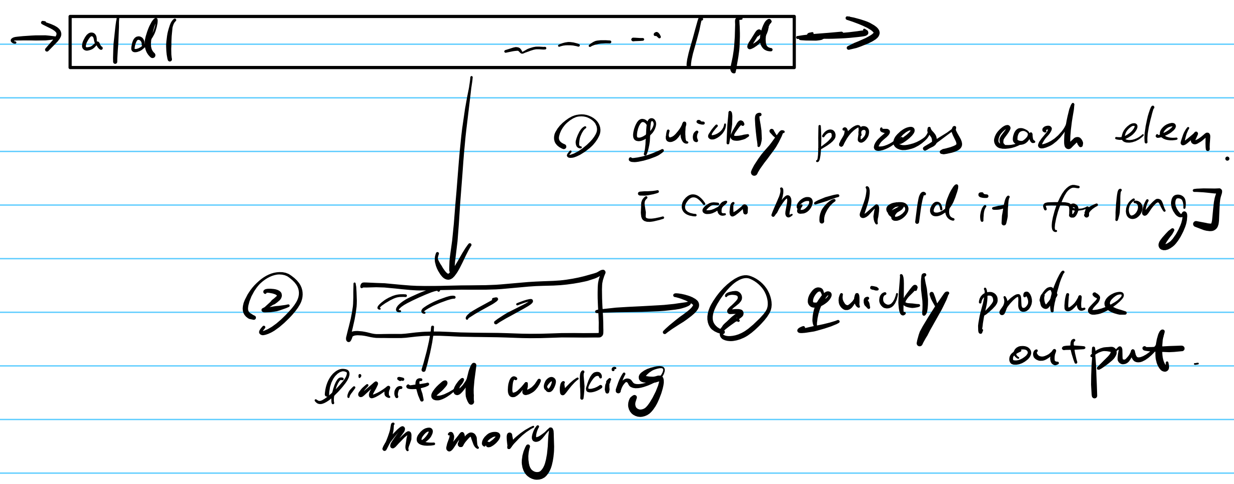
☆ ✓ PATCH 1: amplify succ. probability of (1-sided error) randomized algorithms.

b. Streaming algorithms

- Motivation: internet traffic.



observe packets transmitted



- Data stream Model. [Alon, Matias, Szegedy '96].
  - Model stream as $\underline{m}$ element from $\{1, \ldots, n\}$

    $\langle a_1, \ldots, a_m \rangle = 3, 5, 37 \ldots\ldots$
  - Goal: Compute a function of the stream

e.g. average, median, # distinct elem's.

- Toy example: a stream $\langle a_1, \cdots, a_{n-1} \rangle$
  $n-1$ distinct elem's from $\{1, \cdots, n\}$.
  Goal: find the missing elem. using $O(\log n)$ memory

$$\underbrace{\sum_{k=1}^{n} k}_{O(n^2)} - \sum_{i=1}^{n-1} a_i$$

space to store n^2 : $\log n^2 = O(\log n)$.

- Reservoir Sampling (蓄水池采样)
  - → $\langle a_1, \cdots, a_m \rangle$ m is known (distinct)
    ↓
    Sample an elem s from it unif. at random
    pick a_i w.p $\frac{1}{m}$
  - → what if m is unknown?

$$\boxed{\begin{array}{l} \text{Res. Sampling:} \\ \text{1. init. } s \leftarrow a_1 \\ \text{2. on seeing } t^{th} \text{ element.} \\ \quad \text{update } s \leftarrow a_t \text{ w.p } \frac{1}{t} \\ \quad (\text{unchanged w.p. } (1-\frac{1}{t})) \end{array}}$$

Analysis: At time t $\langle a_1, \cdots, a_t \rangle$

<u>Want</u> $s = a_i$ w.p $\frac{1}{t}$ $\forall i = 1 \cdots t$

Look at a_i $(\forall i, i \leq t)$

$$P[s = a_i] = P[a_i \text{ chosen at tim } i \;\wedge\; i+1 \to t \text{ Not changed}]$$

indep events

$$\begin{aligned} &= P[a_i \text{ chosen at } i] \\ &\quad \cdot P[a_i \text{ NOT changed at } i+1] \\ &\quad \cdot P[\cdots \quad i+2] \\ &\quad \cdot \; \vdots \\ &\quad \cdot P[\cdots \text{ at } t] \end{aligned}$$

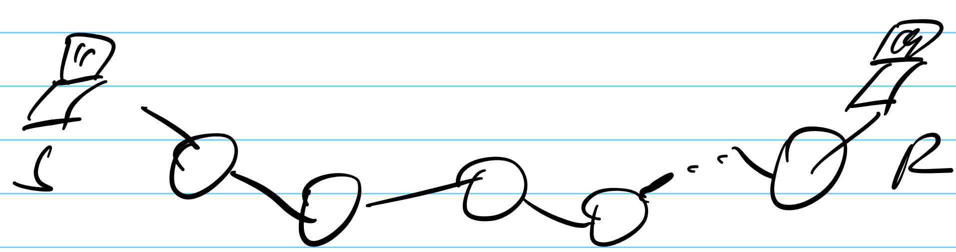
$$= \frac{1}{i}\left(1 - \frac{1}{i+1}\right)\left(1 - \frac{1}{i+2}\right)\cdots\left(1 - \frac{1}{t}\right)$$

$$= \frac{1}{\cancel{i}} \cdot \frac{\cancel{i}}{\cancel{i+1}} \cdot \frac{\cancel{i+1}}{\cancel{i+2}} \cdots \cancel{\cdot} \frac{\cancel{t-1}}{t}$$

$$= \frac{1}{t}$$

#

PATCH 2:



<u>Goal</u>: R wants to learn names of all

routers along the PATH.

<u>limit</u>: each packet can carry
one router info. (& a counter)

<u>Idea</u>: every packet records a <u>random</u> router.

How? → reservoir sampling!

☆ N.B.: P.T. & streaming. → $\boxed{\text{sublinear algorithms}}$
→ massive data
→ long access time

computing useful info. by reading
a <u>tiny portion</u> of data & runs
in <u>sublinear</u> time (wrt full input)
$O(\log n)$

- Randomization is key!