* Introduction, Course Outline

→ Grading

Problem sets → 5% + 3×10%
Project → 15%
Mid-Sem → 20% Sept 16^{th}
End-Sem → 30%

→ Las Vegas → Always correct solutions
→ Monte Carlo → With probability of being correct.

For Lec 2 (Wed 2) to Lec 8 (Fri 11^{th}) — Reter bala notes

Lec 13 — Lec 16 → Ball & Bins to Bloom filter — Intense reading

Lec 19 Bit-probe — In notes (Physical) Lec 21 Cuckoo Hashing pt 1

Min cut in graphs

$G(V, E)$

→ A cut set is a set of edges whose removal disconnects G.

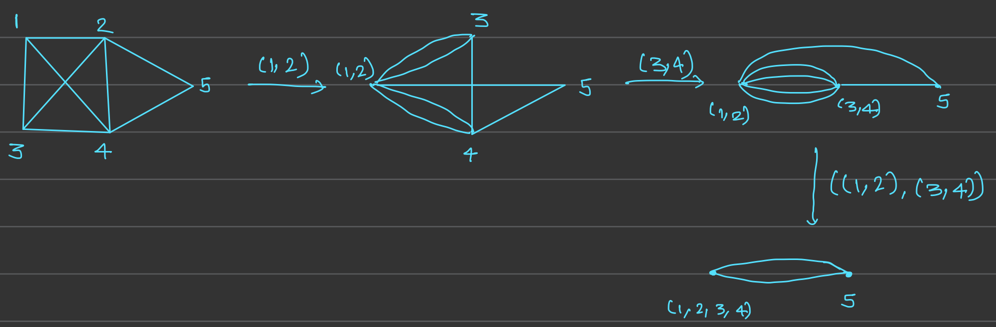
→ Min cut – cut set of minimum cardinality.

→ Karger 93 – Rando Algo

Choose $e \in E$ uar
Contract e
$G \leftarrow G \setminus \{e\}$ & keep parallel edges

} Repeat till 2 vertices

Remaining edges belong to min-cut.



Min-cut = 2

→ If K is size of min cut

$|E| \geq \frac{nK}{2}$ (degree of each vertex $\geq K$)

Fix min-cut c of size k

Pr(No edge from C is sampled) $\geq 1 - 2/n$

E_i → Event where an edge from C was not sampled in the i^{th} step

$$Pr(E_i) \geq 1 - \frac{2}{n}$$

→ Assume E_1 occurred

$G' = G \setminus \{e\}$

Every cut in G' is a cut in G → Min cut size in G' is also k

$F = \bigcap_{j=1}^{i} E_j$ → Event that no edge from C where sampled

$Pr(E_2 | F_1) \geq 1 - \frac{2}{n-1}$ $\quad$ $Pr(E_i | F_{i-1}) \geq 1 - \frac{2}{n-i+1}$

$Pr(F_{n-2}) \geq \frac{2}{n(n-1)}$ ← Pr of final G being min cut

∴ $\theta(n^2)$ mincuts exist.

$Pr(C$ was not output after k times$) \leq \left(1 - \frac{2}{n(n+1)}\right)^k < e^{\frac{-2k}{n(n+1)}}$

Max cuts

↳ NP hard.

→ Partition on V to V_1, V_2 s.t. $|E(V_1, V_2)|$ is maximised

→ α approximation – gives an output s.t size(out) $\geq \alpha$(max cut)

$\alpha = 0.878$

$\alpha = 0.94$ unless $P = NP$

→ We attempt $\alpha = 1/2$ with high probability.

→ Greedy algorithm will give us $\alpha = 1/2$

→ Randomised Algorithm :-

$\exists v \in V$, put v in V_i with prob $1/2$ independent of other variables

- Random variable

$A(x, r)$

$X : \Omega \to \mathbb{R}$ ↪ Cut size. $\quad$ $E(x) \geq$ # of edges$/2$

↳ Set of n bits showing split

10^{th} Aug

* Method of Conditional Expectation

→ X & Y are rvs

$$E(X \mid Y=y) = \sum_{x \in X} x \cdot \Pr(X = x \mid Y = y)$$

→ Facts: ← check book

$$E(x) = \sum_{y} \Pr(Y=y)\, E(X \mid Y = y)$$

$$E\left(\sum_{1}^{n} x_i \mid Y=y\right) = \sum_{1}^{n} E(x_i \mid Y=y)$$ → Try this

$$E(x) = \Pr(v_1 \in V_1) \cdot E(x \mid v \in V_1) + \Pr(v \in V_2)\, E(X \mid v \in V_2)$$
$$= \tfrac{1}{2}\left[E(x \mid v \in V_1) + E(x \mid v \in V_2)\right]$$

Put $v_1 \in V$

$$E(x \mid v_1 \in V_1) = \tfrac{1}{2} E(x \mid v_1 \in V_1 \,\&\, v_2 \in V_1) + \tfrac{1}{2} E(x \mid v_1 \in V \,\&\, v_2 \in V_2)$$

$$\max\left(E(x \mid v_1 \in V_1, v_2 \in V_1),\ E(x \mid v_1 \in V_1, v_2 \in V_2)\right) \geq |E|/2$$

$$E(X \mid v_1 \ldots v_{i-1}) = \Pr(v_1 \in V_1)\, E(x \mid v_1 \ldots v_{i-1}, v_i \in V_1)$$
$$+ \Pr(v_2 \in V_2)\, E(x \mid v_1, v_2 \ldots v_{i-1},$$

⇑ Till here shit notes

14th Aug

* Markov Inequality

If x is non negative RV, $a > 0$

$$\Pr(x \geq a) \leq \frac{E(x)}{a}$$

$$\Pr(x \geq k\,E(x)) \leq 1/k$$

→ Proof

$$I = \begin{cases} 1 & x \geq a \\ 0 & x < a \end{cases} \qquad I \leq X/a$$

$$\Pr(x \geq a) = E(I) \leq E(X/a)$$

$$\Downarrow$$

$$\Pr(x \geq a) \leq E(x)/a$$

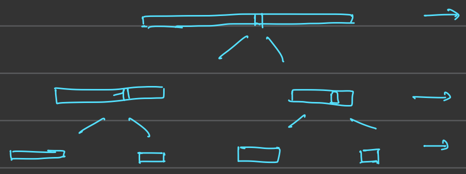
⇒ Quicksort

X = # of comparisions

$$E(x) = 2n\ln n + \theta(n)$$

Theorum $\Pr(x \geq kn\ln n) \leq 1/n$ $\leq \frac{2}{k}$ (MI)

$\log(n)$



Each level has n comparisions.

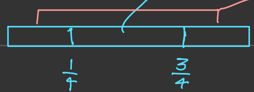
$X_i \rightarrow$ Size of an array on the i^{th} level.

$X_0 = n$

$E(x_i) = \sum Pr(x_{i-1} = t)\ E(x_i / x_{i-1} = t)$

$E(x_i \mid x_{i-1} = t) \leq \frac{1}{2} \cdot \frac{3}{4} t + \frac{1}{2} t = \frac{7}{8} t$ → Other ½ for max size $= t$

→ wp ½ array is partitioned into 2 parts of atmost $\frac{3}{4} t$ size



$$E(x_i) \leq \frac{7}{8} \sum Pr(x_{i-1} = t)\, t = \frac{7}{8} E(x_{i-1}) \leq \left(\frac{7}{8}\right)^2 E(x_{i-2})$$

$$E(x_i) \leq \left(\frac{7}{8}\right)^i n$$

For $i = 3 \log_{\frac{8}{7}} n$ $\qquad E(x_i) \leq \frac{1}{n^2}$

$Pr(x_i \geq 1) \leq \frac{1}{n^2}$ for $i = 3 \log_{8/7} n$

$E_a =$ event that a exists in a subarray inequality of size > 1 after $3 \log_{8/7} n$ recursive steps

$Pr(E_a) \leq \frac{1}{n^2}$ $\qquad Pr(\exists a \text{ st } E_a) \leq \sum Pr(E_a) \leq \frac{1}{n}$

↳ Prob of algo finishing after $3 \log_{8/7} n$ steps

→ $Var(X) = E\left((X - E(X))^2\right) = E(X^2) - E(X)$

→ $Var(X + Y) = Var(X) + Var(Y) + 2E((x - \mu_x) \cdot (y - \mu_y))$

↳ $= 0$ if x, y are independant.

→ $X \rightarrow$ Binomial(n, p)

$Var(X) = Var(x_1 + x_2 + x_3 \cdots x_n) = \sum Var(x_i)$ ↗ Bernoulli(p)

$Var(X_i) = E(X_i^2) - E(X_i) = p - p^2$

$Var(X) = np(1-p)$

$\rightarrow X \rightarrow Geo(p)$

$E(X^2) = Pr(Y=0)\, E(X^2 \mid Y=0) + Pr(Y=1)\, E(X^2 \mid Y=1)$

$= (1-p)\, E((Z-\mu)^2) + p$

$= \frac{2}{p^2} - 1/p$

$Var(X) = \frac{2}{p^2} - \frac{1}{p} - (1/p^2) = \frac{1-p}{p^2}$

Aug 16^{th}

* Chebyshev's Inequality → Extension of MI

→ Let X be a random variable & $a > 0$ then

$Pr[|X-\mu| \geq a] \leq \frac{Var(X)}{a^2}$

→ Proof

$Y = (X-\mu)^2$

$Pr(Y \geq a^2) \leq \frac{E(Y)}{a^2} \rightarrow Pr(|X-\mu| \geq a) \leq Var(X)/a^2$ (Markov)

* Randomness - efficient probability amplification

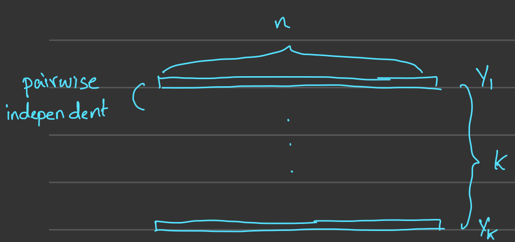
⇒ Frievald's Algorithm

$AB \stackrel{?}{=} C$ $n \times n$, n random bits.

→ if $AB \neq C$ $Pr(\text{Error}) \leq 1/2$
elif $AB = C$ $Pr(\text{Error}) = 0$

- On repeating k times
  $Pr(\text{Error}) \leq 1/2^k$ → nk random bits used

→ $Pr(\text{Error}) \leq 1/k$, # of random bits $= 2n$



→ $n \log k$ random bits sufficient.

Let $Y_1, Y_2 \dots Y_k$ → pairwise independent strings $Pr(Y_i = \bar{b}) = 1/2^n$ uniform.

$$X_i = \begin{cases} 1 & \text{if } ABY_i \neq CY_i \\ 0 & \text{o/w} \end{cases}$$

$E(X_i) \geq 1/2$ $X = \sum_1^k X_i$ $E(X) \geq k/2$ $Var(X_i) \leq 1/4$

Error prob $= Pr(X = 0)$

$$Pr(X=0) \leq Pr(|X - E(X)| \geq k/2) \leq \frac{4\,Var(X)}{k^2} = \frac{4\sum Var(X_i)}{k^2} \leq \frac{1}{k}$$

$Var(\sum X_i) = \sum Var(X_i) + 2 \sum Cov(X_i, X_j)$

$> 2^n$

$\Rightarrow$ Let p be a prime number & sample $a, b \in_{uar} \{0, 1, \dots p-1\}$

$$Y_i = ai + b \pmod p$$

① Y_i is distributed uniformly randomly in $\{0, 1, \dots p-1\}$

② $\{Y_i\}$ are pairwise independent.

$\rightarrow \mathbb{Z}_p = \{0, \dots p-1\}$ $+, \cdot$ $\}$ modulo p $\rightarrow$ Set of numbers with 2 operations

$P \rightarrow$ prime.

Associativity

Commutability

Additive Identity

Multipicative Identity

Add inverse

$\Big\}\ \mathbb{Z}_p$

Mul inverse $- \mathbb{Z}_p - \{0\}$

Finite Field.

Aug 17^{th}

$Y \rightarrow X$

① $\forall c \in \mathbb{Z}_p$

$$Pr[ai + b = c] = 1/p$$

$a, b \in_r \mathbb{Z}_p$

fix a or b; we get a single possible option for the other term.

② $\forall c_1, c_2 \in \mathbb{Z}_p$ $\forall i, j$ $\nearrow \frac{1}{p^2}$

$$Pr[ai + b = c_1 \wedge aj + b = c_2] = \underbrace{Pr[ai + b = c_1] \times Pr[aj + b = c_2]}_{1/p}, \quad a, b \in \mathbb{Z}_p$$

$ai + b = c_1$

$aj + b = c_2$

unique values for a, b for given i, j, c_1, c_2

⇒ Theorum ⇒ For every prime p & integer k $\exists$ a unique finite field of cardinality p^k upto isomorphism.

↳ Galois Field $(GF(p^k))$ → Polynomial k time to create.

↳ We will not see proof.

→ Construction of $GF(2^2)$

Polynomials of degree at most 2 with coefficients $GF(2)$

↳ 8 possible polynomials

degree k over $GF(p)$

$0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1$

000 . . . 011 . . . 110 111

irreducible polynomials of degree 2 over $GF(2)$

x^2+x+1

$$x^2+1 = x^2+1+x+x = (x+1)^2$$

↳ 1 is root over $GF(2)$

For higher values of p, k we get other irreducible polynomial which construct different finite field similar by isomorphism.

$GF(4) = \{0, 1, x, x+1\}$ modulo x^2+x+1, $\{+, \cdot\}$

$x \cdot x \equiv 1+x \pmod{x^2+x+1}$

$x(1+x) \equiv 1 \pmod{x^2+x+1}$ → check this out

$(1+x)(1+x) \equiv x \pmod{x^2+x+1}$

Fact :- $\forall l \geq 0$ $x^{2\cdot 3^l} + x^{3^l} + 1$ is irreducible over $GF(2)$

Aug 18^{th}

* Chernoff & Hoeffding's bounds → Solely for Binomial

$X = \sum_{1}^{n} X_i \qquad X_i = \begin{Bmatrix} 1 & \text{wp } p \\ 0 & \text{wp } 1-p \end{Bmatrix}$, X_i are independent

$\mu = E(X)$

① $Pr(|X-\mu| \geq t) \leq e^{-2t^2/n}$ $\qquad t > 0 \qquad \delta < 1$

② $Pr(X \geq (1+\delta)\mu) \leq e^{-\delta^2\mu/3}$

③ $Pr(X \leq (1-\delta)\mu) \leq e^{-\delta^2\mu/2}$

* Hoeffding's extension

X_i's are independent & take values in $[a, b]$ & $E(X_i) = \mu$

$$Pr\left\{ \left| \frac{1}{n}\sum_{1}^{n} X_i - \mu \right| \geq \epsilon \right\} \leq 2e^{-2n\epsilon^2/(b-a)^2}$$

→ So far all 1-sided algorithms.

For 2-sided algorithm

$Pr(\text{error}) \leq \frac{1}{2} - \epsilon \qquad \epsilon = 1/n^{100}$

↓ reduced to

$1/n^{100}$

Using C&H bounds

For better bounds

on repeating

Repeat algo k times, take majority.

$X_1 \cdots X_k$, $X_i = \begin{Bmatrix} 1 & \text{if } i\text{th iteration is correct} \\ 0 & \text{o/w} \end{Bmatrix}$ $\quad E(X_i) \geq \frac{1}{2} + \epsilon$

$X = \sum_{1}^{n} X_i \Rightarrow E(X) \geq k/2 + k\epsilon$

$$Pr(\text{Error}) = Pr(X < k/2)$$

$$Pr(E(X) - X > k\epsilon) \le e^{-2k\epsilon^2} \qquad k = \frac{n}{\epsilon} \qquad P(\text{Error}) \le e^{-n}$$

* Moment generating function

$$M_X(t) = E(e^{tX})$$

$$= E\left(\sum \frac{t^i X^i}{i!}\right)$$

$$= \sum_{i \ge 0} \frac{t^i}{i!} E(X^i)$$

↳ i^{th} moment of X

$$Pr(X \ge m) = Pr(e^{tX} \ge e^{tm}) \le \frac{E(e^{tX})}{e^{tm}} \qquad \text{(Markov)}$$

What if k wise independent.

$$E(e^{tX}) = E(e^{t\sum X_i}) = E\left(\prod_1^n e^{tX_i}\right) = \prod_1^n E(e^{tX_i}) \qquad \text{(Independent)}$$

$$E(e^{tX_i}) = pe^t + (1-p) \qquad (1-p) = q$$

1 $\le e^{p(e^t - 1)}$

$$(pe^t + q)^n \le e^{pn(e^t - 1)} \qquad pn = \mu$$

$$Pr(X \ge m) \le \frac{(pe^t + q)^n}{e^{tm}} \qquad m = (p+r)n \qquad \mu = pn$$

$pn = \mu$ $\quad r/p = \delta$

$$Pr(X \ge (p+r)n) \le \frac{(pe^t + q)^n}{e^{t(p+r)n}}$$

2 $$Pr(X \ge (1+\delta)\mu) \le \frac{e^{\mu(e^t - 1)}}{e^{t\mu(1+\delta)}}$$

$$\le \left(\left(\frac{p}{p+r}\right)^{p+r} \left(\frac{q}{q-r}\right)^{q-r}\right)^n$$

$$\le \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu} \le e^{\frac{-\delta^2\mu}{3}}$$

→ k servers, n jobs

Allocate jobs to servers in a decentralized way

Naive : Randomly apply given job to any server.

Expected Load = n/k — Balls in Bins process ⇓ Start of Balls & Bins

Aug 21 → Read up to clear misunderstandings.

→ m balls, n bins

Throw balls into bins uar.

→ Coupon Collector — What is the value of m st every bin contains atleast one ball.

→ Hashing — How large should n be compared to m so that no bin contains a lot of balls.

→ Birthday problem — What should m be so that the prob ∃ bin with more than one ball ≥ ½

B_i — i^{th} ball lands in a bin of its own.

$$\Pr\left(\bigcap_1^m B_i\right) = \left(1 - \frac{i-1}{n}\right) \Pr\left(\bigcap_1^{m-1} B_i\right)$$

$$= \prod_1^m \left(1 - \frac{i-1}{n}\right)$$

$$Pr\left(\bigcap_{1}^{m} B_i\right) \leq \prod_{1}^{m} e^{-(i-1)/n}$$

$$\leq e^{-m(m+1)/2n} \quad < \frac{1}{2}$$

$$m = \Theta(\sqrt{n})$$

→ E_i – event that i^{th} ball does not land in a seperate bin given that first $i-1$ all landed in seperate bins

$$Pr(E_i) = {}^{i-1}/_n$$

$$Pr\left(\bigcup_{1}^{m} E_i\right) \leq \sum_{1}^{m} \frac{i-1}{n} = \frac{m(m-1)}{2n}$$

if $m < \sqrt{n}$, $\exists$ a ball that doesn't land in a seperate bin with probability $< \frac{1}{2}$

→ If n balls are thrown into n bins, what is the maximum load ?
– $\Theta(\ln n / \ln \ln n)$ wp $(1 - 1/n)$

→ Theorum – If n balls are thrown into n bins uar, then wp $\geq 1 - 1/n$,

$$\text{max load} \leq \frac{3 \ln n}{\ln \ln n}$$

Proof :

$$Pr(\text{bin } i \text{ has atleast } k \text{ balls}) \leq \binom{n}{k} \left(\frac{1}{n}\right)^k$$

$$\leq \left(\frac{ne}{k}\right)^k \frac{1}{n^k} = \left(\frac{e}{k}\right)^k$$

$$Pr(\exists \text{ bin } i \text{ with atleast } k \text{ balls}) \leq n \left(\frac{e}{k}\right)^k \leq 1/n$$

→ $X_1, X_2 \; . \; . \; . \; X_n$ → # of balls in individual bins for m balls n bins

$X_i \rightarrow$ binomial $(m, 1/n)$

$$Pr(X_i = k) = \binom{m}{k} (1/n)^k (1 - 1/n)^{m-k}$$

$$= \frac{m!}{k!\,(m-k)!} \cdot \frac{1}{n^m} (n-1)^{m-k}$$

$$\simeq \frac{1}{k!} \left(\frac{m}{n}\right)^k e^{-m/n} \qquad m, n >>> k$$

→ Poisson's Distribution (λ)

$X \sim$ Poisson (λ) → $Pr(X = k) = \dfrac{e^{-\lambda} \lambda^k}{k}$

Theorum : if $X \sim Bin(n, p)$ & $\lim_{n \to \infty} np = \lambda$

for every fixed k

$$\lim_{n \to \infty} Pr(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

Missing 23 Aug

Aug 24

→ $\{X_i^m\}_{1\le i\le n}$ — dist of balls in bins   m balls   n bins

$\{Y_i^m\}_{1\le i\le n}$ $Y_i^m \sim \text{Poisson}(m/n)$, independent.

$f \to$ non-negative function

$$E(f(X_1^m, \dots X_n^m)) \le e\sqrt{m}\, E(f(Y_1^m, \dots Y_n^m))$$

→ Proof

$$E(f(Y_1^m, Y_2^m \dots Y_n^m)) = \sum E(f(Y_1^m \dots Y_n^m) \mid \textstyle\sum Y_i = k) \cdot \Pr(\textstyle\sum Y_i = k)$$

$$\ge E(f(Y_1^m \dots Y_n^m) \mid \textstyle\sum Y_i = m) \cdot \Pr(\textstyle\sum Y_i = m))$$

$m! < \sqrt{2\pi m}\left(\frac{m}{e}\right)^m$

Stirling's Approx →

$$\ge E(f(X_1^m \dots X_n^m)) \cdot \frac{e^{-m} m^m}{m!}$$

$$\ge E(f(X_1^m \dots X_n^m))$$

→ Theorum

If n balls are thrown into n bins uar then wp $\ge 1 - 1/n$, $\exists$ bin containing $\Omega\left(\frac{\ln n}{\ln \ln n}\right)$ balls.

→ Proof

$$f(x_1, \dots x_n) = \begin{cases} 1 & \text{if } \max(x_1, \dots x_n) < k \\ 0 & \text{o/w} \end{cases}$$

$$E(f(x_1^n \ldots x_m^n)) = Pr(\text{every bin} < k \text{ balls})$$

$$Pr(Y_1^n \geq k) \geq Pr(Y_1^n = k) = \frac{1}{ek!}$$

$$Pr\left(\bigcap_1^m (Y_i^n < k)\right) \leq \left(1 - \frac{1}{ek!}\right)^n \leq e^{\frac{-n}{ek!}}$$

Missing Bloom Filters

shit notes

⇑

Aug 28^{th}

* Universal Hash Family (Carter Wegman)

→ A family $H = \{h : U \to [n]\}$ is said to be k-universal if $\forall$ distinct $x_1, \dots x_k \in U$

$$\Pr_{h \in H}\left[h(x_1) = h(x_2) \dots h(x_k)\right] \leq \frac{1}{n^{k-1}}$$

⇒ Strongly universal / k-wise ind

→ $H = \{h : U \to [n]\}$ is " if $\forall$ distinct $x_1 \dots x_k$ & $y_1 \dots y_k \in [n]$

$$\Pr\left[\bigcap_1^k h(x_i) = y_i\right] = \frac{1}{n^k}$$

(i) $\Pr_{h \in_r H}[h(x) = y] = \frac{1}{n}$ (ii) $\Pr_{h \in_r H}\left[\bigcap_1^k h(x_i) = y_i\right] = \prod_1^k \Pr(h(x_i) = y_i)$

→ Strongly universal → Universal Hash family $(p \to q)$ $(\cancel{q \to p})$

→ m balls into n bins using 2-universal hash family
$H = \{h : [m] \to [n]\}$

Choose $h \in_r H$
Assign balls into bins using h.

→ $X_{ij} = \begin{cases} 1 & i,j \to \text{same bin} \\ 0 & \text{o/w} \end{cases}$ $E[X_{ij}] = \Pr[h(i) = h(j)]$

→ For balls i & j $\Pr(X_{ij} = 1) \leq 1/n$

→ Total number of collisions $= X = \sum_{i<j} X_{ij}$ $E(X) \leq \binom{m}{2} \cdot 1/n \leq \frac{m^2}{2n}$

→ Let Y be the max load $\binom{Y}{2} \leq X$

↳ # of collisions in max load bin

$\Pr\left(X \geq \frac{m^2}{n}\right) \leq 1/2$

$\Pr\left(\binom{Y}{2} \geq \frac{m^2}{n}\right) \leq 1/2$ $\frac{(Y-1)^2}{2} \leq \binom{Y}{2} \leq \frac{Y^2}{2}$

$\Pr\left(Y \geq 1 + m\sqrt{\frac{2}{n}}\right) \leq 1/2$

If $m = \theta(n^2)$ → w.p $\geq 1/2$ every bin has $\theta(1)$ balls.

→ Static Dictionary $S \subseteq U$, $|S| = m$, we take $n = \theta(m^2)$
Sample h from $\mathcal{H}$, with expected # 2 runs, we can sample h with every bin of load $= 1$

→ For $n = \theta(m)$ → FKS hashing (coming up)

$\Rightarrow$ Dynamic Dictionary

- additions
- deletions
- query

$\rightarrow$ Set $S \subseteq U$ $\curvearrowright$ set of all instructions

$x \in U \quad h \in \mathcal{H}$ (2-universal)

$$\mathrm{coll}(x, S, h) = |\{y \in S \mid h(x) = h(y)\}|$$

$$\underset{h \in_r H}{E}(\mathrm{coll}(x, S, h)) = \frac{1}{|\mathcal{H}|} \sum \mathrm{coll}(x, S, h)$$

$\curvearrowright$ indicator function

$$= 1/|\mathcal{H}| \sum_{h \in \mathcal{H}} \sum_{y \in S} 1_{[h(x) = h(y)]}$$

$$= \sum_{y \in S} \underbrace{\frac{1}{|\mathcal{H}|} \sum_{h \in_r H} 1_{[h(x) = h(y)]}}_{\underset{h \in_r H}{\Pr}[h(x) = h(y)] \leq 1/n}$$

$$\leq |S|/n$$

⇒ Explicit 2 - universal family

$U, |U| = m \longrightarrow n$

prime $p > m$ $\quad \mathcal{H} = \{ h_{a,b} \mid 1 \leq a \leq p-1,\ 0 \leq b \leq p-1 \}$

$h_{a,b}(x) = ((ax+b) \bmod p) \bmod n \qquad |\mathcal{H}| = \Theta(p^2)$

$\Pr_{h \in_r H} [h(x) = h(y)] \leq 1/n$

$$\left.\begin{array}{l} ax + b = z_1 \pmod p \\ ay + b = z_2 \pmod p \\ z_1 = z_2 \pmod n \end{array}\right\} \rightarrow \text{To be done on Wednesday}$$

Aug 30^{th}

For every $z_1 \neq z_2$, $\exists$ unique solⁿ for a, b

$|z_2 \mid z_2 \neq z_1,\ z_1 = z_2 \pmod n| \leq p/n - 1$

$\Pr[h(z_1) = h(z_2)] \leq \dfrac{p(p/n - 1)}{p-1} \leq \dfrac{1}{n}$ → check

⇒ Pairwise independent Hash family

$|U| = 2^m \longrightarrow 2^n$ → verify.

field $GF(2^m)$ $\qquad \mathcal{H} = \{ h_{a,b} \mid a, b \in GF(2^m) \}$

$h_{ab}(x) = \underbrace{ax + b}_{\text{truncate to } n \text{ bits.}}$ in $GF(2^m)$

↳ m bit

For k-wise independent hash family $\qquad a_1 x^{k-1} + a_2 x^{k-2} \cdots a_{k-1} x + b$ in $GF(2^m)$

⇒ Perfect hash family

→ A family $\mathcal{H}$ of functions $\{h : \mathcal{U} \to [n]\}$ is perfect for sets of size $\leq n$ if $\forall S \in \mathcal{U}$, $|S| \leq n$ $\exists h$ st $\forall x \neq y$ $h(x) \neq h(y)$

→ Implies that Static Dict has $O(n)$ data structure with $O(1)$ query time.

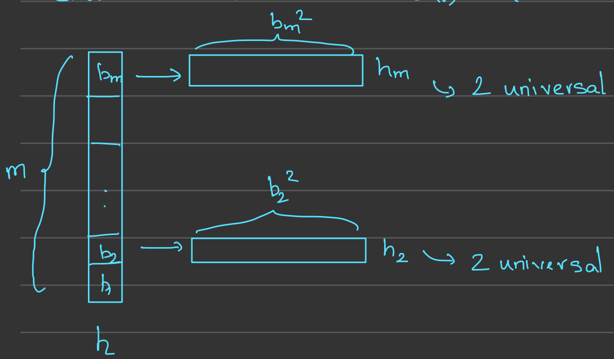
→ Each cell on the table $= O(\log |\mathcal{U}|)$ bits (each stores $|\mathcal{U}|$ data)
$|\mathcal{H}| = |\mathcal{U}|^{O(1)}$

→ If $\mathcal{H} = \{h : |\mathcal{U}| \to [n]\}$ is a perfect hash family for sets of size n, then
$|\mathcal{H}| = 2^{\Omega(n)}$ → try if not in notes.

⇒ FKS hashing

$\mathcal{U} \to [n]$ $|S| = m$

Choose $h \in \mathcal{H}$ (2-universal) s.t # of coll $\leq m$



$$\text{Size} = m + \sum_{1}^{m} b_i^2 = O(m)$$

$$\sum_{i=1}^{m} \binom{b_i}{2} = \text{total \# of collisions} \leq m$$

$\Rightarrow$ Bit probe model

$\rightarrow$ Size = # of bits $\rightarrow$ $O(m \log |U|)$

query = # of bits that are probed. $\rightarrow$ $O(\log |U|)$

Sep 1st Fri

* Problem Set.

1) Generate n distinct #s for each $i \in [n]$ and sort according the generated numbers

For each $i \rightarrow 3\log n$ bits

$Pr(k_i = k_j) = \frac{1}{2^{3\log n}} = \frac{1}{n^3}$

$Pr(\exists i \; k_i = k_j) \leq 1/2$

$Pr(\forall i \; k_1 \neq k_j) \geq 1/2$ $\Theta(n \log n)$

Q4: Find variance of Y

Q5

$\bar{x}_1, \bar{x}_2 \ldots \bar{x}_n \in \{0,1\}^l$

$x_i = \sum_{1}^{l} x_{ij} y_j \pmod 2$ for $\bar{y} = \{0,1\}^l$ uar

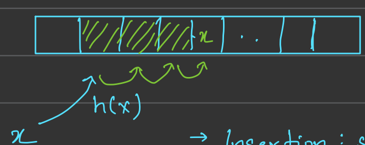
$\Pr(x_{i_1} = b_1, x_{i_2} = b_2 \ldots x_{i_k} = b_k) = 1/2^k$ for $x_{i_1}, x_{i_2} \ldots x_{i_k}$

$$\begin{bmatrix} x_{11} & \cdots & x_{1l} \\ & & \\ x_{k1} & & x_{kl} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}$$

Sep 3rd Wed

$\Rightarrow$ Open Addressing - Linear probing.

→ No auxillary data structures



→ Insertion : starts at $h(x)$, insert at first available slot.

→ Membership : Start at $h(x)$, continue till you find x or an empty slot.

→ Doesn't work with deletions, we can keep a flag (tombstone) and change above.

→ Works well in practice as cache miss brings an entire block into cache.

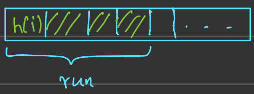
⇒ Knuth '63

If h is completely random, then expected time is $O(1)$.

→ 2010 – 11   5 – wise Ind, is necessary & sufficient for expected $O(1)$ time.

→ Run : A maximal contigous sequence of cells that are filled.

Given $i \in S$, length of the run containing $h(i)$



→ k - wise independent family H $h \in_r H$

$$I_j = [h(i) - (2^j - 1),\ h(i) + (2^j - 1)] \rightarrow \text{dyadic intervals}$$

↳ Interval of positions in the hash table

→ $|S| = n$, hash table has size : t

→ $E(\#\text{ of elements in } I_j) = \frac{|I_j|}{t} \cdot n$

$$X_i = \begin{cases} 1 & \text{if } i \in I_j \\ 0 & \text{o/w} \end{cases} \qquad X = \sum_{i \in S - \{1\}} X_i \quad \rightarrow E(X) = \frac{(n-1)|I_j|}{t}$$

$|I_j| = 2^{j+1} - 1$ $t = 8n$ $E(x) \leq 2^{j-2}$

⇑

→ $R \to$ run containing 1

$$E(|R|) = \sum_{0}^{n} \ell \cdot Pr(|R| = \ell) \leq \sum_{j=0}^{last} 2^j \, Pr(2^{j-1} \leq |R| \leq 2^j)$$

If $|R| \geq 2^{j-1} \to$ # of elements in I_j is $\geq 2^{j-1}$

→ If h is completely random

$$Pr(|R| \geq 2^{j-1}) \leq Pr(X \geq 2^{j-1}) \quad \to \text{We use chernoff.}$$
$$\leq e^{-2^{j-2}/2}$$

$$E(|R|) \leq \sum_{j=0}^{last} \frac{2^j}{e^{2^j}} = O(1)$$

∴ $h \to$ completely random, Knuff '63 proved.

Summation of $x/e^x \to O(1)$

→ If $h \to$ 3 wise independent hash family.

Chernoff bounds no longer hold.

→ Given $h(1)$ $[h(i)]$, rest are pairwise independent.

$$Var(X) = \sum_{i \in S/\{1\}} Var(X_i) = \sum_i E(X_i^2) - E(X_i)^2 \leq E(X)$$

$$Pr[|X - \mu| \geq a] \leq \frac{Var(X)}{a^2} \to \text{Chebyshev}$$

Do this !!

$$Pr(X \geq 2^{j-1}) \leq \frac{Var(X)}{}$$

$E(|R|) = O(\log n)$

→ If h – 5 wise,

given $h(1)$, rest are 4 – wise, we need a 4 wise bound.

→ $X_1, \dots X_n$ → 4 wise independent s.t $E(X_i) = p$ & Bernoulli

$X = \sum X_i \quad E(X) = np = \mu$

For $\mu \geq 1$ & $\beta > \mu$

$Pr(X > \mu + \beta) \leq \frac{4\mu^2}{\beta^4}$ Try on your own.

$$Pr(|R| \geq 2^{j-1}) \leq Pr(X \geq 2^{j-1}) \leq Pr(X \geq 2^{j-2} + 2^{j-2})$$

$$\leq \frac{4 \; 2^{2(j-2)}}{2^{4(j-2)}} = \frac{32}{2^{2j}}$$

$$E(|R|) \leq \sum_{0}^{last} 2^j \cdot \frac{32}{2^{2j}} \leq \sum_{0} \frac{\theta(1)}{2^j} = \theta(1)$$

* Cuckoo Hashing, till all possibilities

→ unicyclic → Atmost 1 cycle

Thu 7^{th} Sep

→ Bound size of connected components

→ Random graph $G(n, m)$ n vertices, m edges ; show

(i) whp, the components are small

(ii) whp, the small components are unicyclic.

Repeat m times

Choose $v_1, v_2 \in V$ & add (v_1, v_2)

⇒ Erdős - Renyi Random graph

$G(n, p)$, for every pair of vertices (v_1, v_2)

add (v_1, v_2) edge wp p independantly.

→ $E(\#\text{ of edges}) = \binom{n}{2} \cdot p$

→ If $p = m / \binom{n}{2}$

$E(\#\text{ of edges}) = m$

⇒ Monotone graph property

ex: $\exists$ con comp with $|C| > k$

→ A property is monotone increasing if $G \in P \wedge G \subseteq G' \Rightarrow G' \in P$

" monotone decreasing → " $\wedge\ G' \subseteq G$ ex bipartiteness, planarity

→ Lemma : Let P be any monotone - increasing property

$$P(n, m) = \Pr_{G \sim G(n,m)}(G \in P)$$

$$P(n, p) = \Pr_{G \sim G(n,p)}(G \in P)$$

• If $p^+ = (1+\epsilon)\frac{m}{\binom{n}{2}}$ & $p^- = (1-\epsilon)\frac{m}{\binom{n}{2}}$ ⇓

$$* \quad P(n, p^-) - e^{-O(m)} \leq P(n, m) \leq P(n, p^+) + e^{-O(m)}$$

$\Rightarrow$ Theorum

Let G be a cuckoo graph with $m = (1-\epsilon)\, n/2$

(i) wp $\geq 1 - 1/n$, every component has size $O(\log n)$

(ii) E(size of each component) $= O(1)$

(i) $G(n, p^+)$ $\quad p^+ = (1+\epsilon)\, m/\binom{n}{2} = (1-\epsilon^2)/n-1$

$\rightarrow$ Start from a vertex, do a bfs to get component size.

Fix $v \in V$, start a BFS

$v_1 \quad |N(v_1)| \sim Bin(n-1, p^+) = N_1$

$v_1, v_2 \cdots v_i \quad - Bin(n-i+1, p^+)$

$\hookrightarrow N_i$

$$Pr\left(\sum_1^k N_i \geq k\right) \leq \underbrace{Pr\left(\sum_1^k B_i \geq k\right) \quad B_i \sim Bin(n-1, p^+)}_{\text{Branching process}}$$

$\sum_1^k B_i = Bin(k(n-1), p^+)$

$E(B) = k(n-1)\, p^+ = (1-\epsilon^2)\, k$

$Pr(B > k) = Pr\left(B > \frac{E(B)}{(1-\epsilon^2)}\right) = Pr(B > (1+\epsilon^2)\, E(B))$

$\hookrightarrow$ Bound w Chernoff

Fri 8^{th} Sep

(i) If $m = (1-\epsilon)\, n/2$, then expected size of any cc is $O(1)$

(ii) whp all components of size $O(\log n)$ are unicyclic.

(i) Fix v

Let S be the size of component in $G(n, m)$

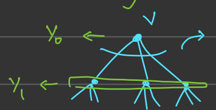
$$E(S) = \sum k \Pr(S = k)$$

$$= \begin{bmatrix} \Pr(S=1) \\ \Pr(S=2) \\ \\ \\ \end{bmatrix} + \begin{bmatrix} \Pr(S=2) \\ \\ \\ \\ \end{bmatrix} \begin{matrix} \\ \cdot \\ \vdots \end{matrix}$$

$$= \sum_{k \geq 0} \Pr(S \geq k) \qquad = \sum_{k \geq 0} \Pr(S' \geq k) + k e^{-O(m)}$$

↳ cc size in $G(n, p^+)$

⇒ Branching process



$\curvearrowright$ $Bin(n-1, p^+)$

→ Each node spawns k children $\sim Bin(n-1, p^+)$, ∴ we bound size of cc.

$$p^+ = \frac{(1-\epsilon^2)}{n-1}$$

→ Let Y_i be # of nodes at level i, $Y_0 = 1$

$$Y = \sum_{i \geq 0} Y_i$$

$$\rightarrow E(Y_i) = \sum \overbrace{E(Y_i \mid Y_{i-1} = k)}^{k(n-1)p^+} \Pr(Y_{i-1} = k)$$

$$E(Y_i) = \sum k(n-1)p^+ \cdot \Pr(Y_{i-1} = k)$$

$$= (n-1)p^+ E(Y_{i-1})$$

$$= ((n-1)p^+)^i = (1-\epsilon^2)^i$$

$E(Y) = \sum_{i \geq 0} (1-\epsilon^2)^i = 1/\epsilon^2 = O(1)$

(ii)

Fix k vertices $\binom{n}{k}$

$\rightarrow$ Cayley's formula : there are k^{k-2} labelled trees on k vertices.

$\binom{m}{k-1} (k-1)! \left(\frac{1}{n^2}\right)^{k-1}$

!!

Add 2 edges , $\binom{m-k+1}{2} \cdot 2! \, (k^2)^2 \left(\frac{1}{n^2}\right)^2$

$\left(1 - \frac{k(n-k)}{n^2}\right)^{m-k-1}$ $\rightarrow$ No edges connecting cc to remaining vertices

$\leq \binom{n}{k} k^{k-2} \binom{m}{k-1} (k-1)! \left(\frac{1}{n^2}\right)^{k-1} \binom{m-k+1}{2} 2! \, k^4 \left(\frac{1}{n^2}\right)^2 \left(1 - \frac{k(n-k)}{n^2}\right)^{m-k+1}$

$\hookrightarrow$ Solve to val

$\Rightarrow$ Midsem Info

9 – 11 AM

2 sides of A4 sheet worth of formulae

END OF MIDSEM

11th Sep Monday

Online Algorithms

⟹ Paging Problem

→ Cache $\boxed{|||||}$

→ Page requests $r_1, r_2, r_3 \cdots r_m$

→ If r_i not in cache (page fault / cache miss), eject some item & bring r_i in.

Goal : Minimize page faults

→ Deterministic Algos :

(i) LRU (ii) FIFO (iii) LFU (frequently)

→ We know entire sequence

→ Optimal offline strat

Farthest in the Future (Kleinberg - Tardos proof)

→ Change goal : Competitive ratio

an algorithm A is c competitive if $\forall$ sequence of requests $r_1, r_2 \ldots r_m$

- $f_A(r_1, r_2 \ldots r_m) \leq c\, f_{OPT}(r_1, r_2 \ldots r_m)$

  ↳ optimal algo (possibly offline)

- $\dfrac{f_A(r_1, \ldots r_m)}{f_{OPT}(r_1, \ldots r_m)}$ is bound.

→ Farthest in the future

$\boxed{1 | 2 | 3 | 4}$ 5 2 3 2 4 1 6 4 4 2 4 3 3 1

$(x, y) \to y$ supplants x ← (1,5) (5,1) (1,6) (6,1) → 4 CMs

→ LRU

5 2 3 2 4 1 6 4 4 2 4 3 3 1

(1,5) (5,1) (3,6) (1,3) (6,1) → 5 CMs

⇒ Theorum : LRU is k-competive → k → cache size

For any sequence, if LRU makes n faults, then FIF makes $\geq n/k$ faults.

→ Divide requests in to phases where LRU makes k faults in each phase.
↳ $(r_1 \ldots r_m)$

$\underbrace{r_1 \ldots r_{i_1}}_{k \text{ faults}} \quad \underbrace{r_{i_1+1} \ldots r_{i_2}}_{k \text{ faults}} \quad \underbrace{r_{i_2+1} \ldots r_{i_3}}_{k \text{ faults}} \quad \ldots \quad \underbrace{r_{i_k+1} \ldots r_n}$

show each phase, FIF has ≥ 1 faults

→ In phase i

(i) Among the k faults, the page r_j faulted twice.

$\geq k+1$ distinct elements

$(\underset{\text{fault}}{r_j} \underbrace{\qquad\qquad}_{\# \text{ of distinct elements } \geq k-1} \underset{\text{fault}}{\overset{r_j \text{ evict}}{r_n}}) \quad \underset{\text{fault}}{r_j}$

→ As cache size is k, atleast 1 page fault by FIF/OPT

(ii) All the k faults are used

r_{i-1} $\underbrace{r_i \quad\quad\quad\quad r_n}_{k \text{ distinct faults}}$

doubt!

(a) r_{i-1} is one of the faults → Case (i)

(b) r_{i-1} is not one of the faults

↳ r_{i-1} lies in cache before phase with k distinct elements being added. At least 1 page fault.

⇒ Theorem: If a deterministic paging algo is c-competitive, then $c \geq k$

Fix an algo A

Construct a sequence s.t A makes n faults, then OPT will make at most n/k faults.

val evicted by $k+1$; phase of k distinct elem

$\boxed{1 | 2 | 3 | \cdots | k}$ $\quad (1 \; 2 \; \ldots \; k)\ (\underbrace{k+1 \;\; e_1 \;\; e_2 \;\; e_3 \;\; \cdot}) \;\cdots\; e_m)$

$e_i \in \{1, 2 \cdots k+1\}$

e_{i+1} = val evicted by e_i

For opt, make phases of k distinct elements, that has 1 page faults

13th Sep Wednesday

⟹ Randomized Paging

→ Oblivious adversary Knows only the source code of the algorithm
Unaware of random bits chosen

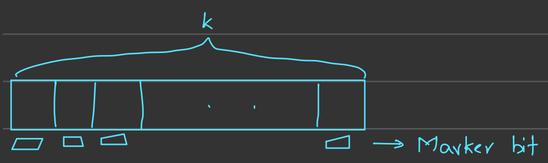
Non adaptive adversary

→ Adaptive adversary
(i) offline adversary
(ii) Online adversary

→ Competitive ratio
An algo A has comp ratio c if $\forall\ r_1, r_2 \ldots r_m$

$$E(f_A(r_1, \ldots r_m)) \leq c\, f_{OPT}(r_1, \ldots r_m)$$

⟹ Marker Algorithm



→ Start of a round – all marker bits are 0
for each req r_i
  if r_i in cache
    set marker bit 1
  if r_i not in cache
    Choose a mb = 0 uar
    place r_i there and set mb = 1

When all bits = 1

⇒ Theorem : Marker is $2H_k$ competitive

$H_k \rightarrow k^{th}$ Harmonic sum

Proof :

Start of Round i

Stale items : Items requested in Round $i-1$ and still requested in round i

New items : Items requested newly in Round i $\rightarrow n_i$

OPT Algo

→ S_i items in cache for OPT

S_i' items in cache for Marker

$d_i = |S_i \setminus S_i'|$

OPT will fault on atleast $n_i - d_i$ ☆

$d_{i+1} = |S_{i+1} \setminus S'_{i+1}|$

OPT will faults on $\geq d_{i+1}$ ☆

→ OPT faults on $\geq \max(n_i - d_i, d_{i+1})$

$$\geq \frac{n_i - d_i + d_{i+1}}{2}$$

→ Over m rounds

$\#$ of faults $\geq \frac{1}{2} \sum_{1}^{m} n_i$

→ Now we need to obtain an upper bound for Marker

$\leq H_k \sum_{1}^{m} n_i$

Marker Algo

→ All new elements will fault

→ Stale elements might fault

→ For worst case scenario

$\Rightarrow$ All n_i elements come first

→ Prob(First stale element is hit) $\leq n_i/k$

→ Second stale element $\leq n_i/k-1$ (One more marker bit is set)

$\therefore$ # of faults $\leq n_i + \frac{n_i}{k} + \frac{n_i}{k-1} \dots \leq n_i(H_k)$ per round

Σ faults $\leq \sum_i^m n_i \, (H_k)$

Thu 14/09 → Missed class

$\Rightarrow$ Yao's Min max principle

→ std technique to lower bound rand algos

Like a game between algo designs & adversary

i–

→ Example 2 player zero sum game

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Entry :- How much col player pays row player

→ Lets take a diff version

	R	P	S
R	0	1	2
P	-1	0	5
S	-2	-1	0

Safest option for row player → R guarantees 0 payoff
P & S could lead to −1, −2

Safest option for col player → R

Row player ⇒ $\max_i \min_j M_{ij}$

Col player ⇒ $\min_j \max_i M_{ij}$

→ If $\max_i \min_j M_{ij} = \min_j \max_i M_{ij}$, game has a value

→ For simple RPS $\max_i \min_j M_{ij} = -1 \neq 1 = \min_j \max_i M_{ij}$
↳ Doesn't have a value

$\forall M \quad \max_i \min_j M_{ij} \leq \min_j \max_i M_{ij}$

⇒ Mixed strategies

→ Action isn't fixed, chosen wp
$p \sim$ distribution over rows $\qquad q \sim$ distribution over columns

→ $E(\text{payoff}) = \sum_{i=1}^{m} \sum_{j=1}^{n} p_i q_j M_{ij} = p^T M q$

→ Row player : $\max_p \min_q p^T M q$

→ Col player : $\min_q \max_p p^T M q$

⇒ Von Neumann minimax theorem

→ For every payoff matrix M (2 person zero sum)

$$\max_p \min_q p^T M q = \min_q \max_p p^T M q$$

→ Suppose we fix p

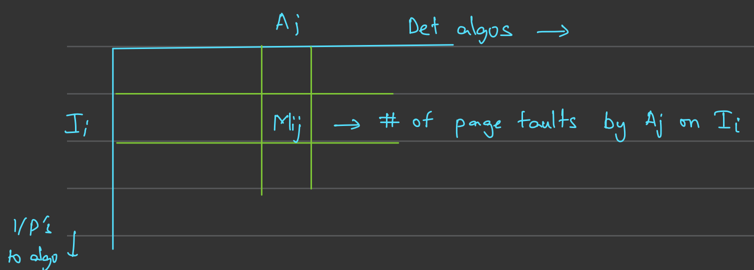
$$p^T M = [\alpha_1, \alpha_2 \cdots \alpha_n]$$

We want to find q which minimizes $p^T M q$ s.t $\sum_i^n q = 1$

$$p^T M q = \sum_i^n \alpha_i q_i \geq \min_i \alpha_i \sum q_i = \min_i \alpha_i \Rightarrow \text{Choose minimum}$$

⇒ Corollary

$$\forall M \quad \max_p \min_j p^T M e_j = \min_q \max_i e_i^T M q$$



We want to show

(For every rand algo) ~ dbn over A

$\exists i$ (some bad input) s.t

$E_q(c(A, I_i))$ is large ⇒ Some row "sum" is large

It suffices to show that all col "sums" are large

⇒ The principle

$\forall\, M, p, q \quad \min_j p^T M e_j \leq \max_i e_i^T M q$

exp cost of det alg when i/p is sampled acc to p

⇓

Finding worst case i/p dbn ∀ det algos

expected cost of a rand algo

⇓

We want to show this is large

* 15th Sep, Fri

⇒ What is to be shown

$\forall$ randomized A, $\exists$ input I st $E(c(A, I)) \geq H_K$
↳ # of page faults

⇒ Corollary

→ $\max\limits_{p} \min\limits_{j} p^T M e_j = \min\limits_{q} \max\limits_{i} e_i^T M q$ ↗ $E(c(A, I_i))$

→ $\min\limits_{j} p^T M e_j \leq \max\limits_{i} e_i^T M q$ → Yao's min max

→ Goal : $\max\limits_{i} e_i^T M q \geq H_K$

Sufficient :

$$\min_{j} p^T M e_j \geq H_K$$

$\exists$ distribution p over the inputs
s.t $E(c(A, I)) \geq H_K$
$I \sim p$
$\forall$ determined Algo A

⇒ Theorem : For any c competitive online paging algorithm
$c \geq H_K$

$\boxed{1 \mid 2 \mid 3 \mid \cdot \quad \cdot \quad \cdot \mid k}$ $\{k+1\}$

$r_1 = K+1$ $r_2 =$ choose uar $\{1, 2, \cdots K+1\}$

$r_i =$ choose uar $\{1, 2, \cdots K+1\} \setminus \{r_{i-1}\}$

→ Divide requests into rounds.

Each round – k distinct page requests

OPT misses at most once every round – throw out element not requested during the round.

→ Fix a det algorithm A

Each A is charecterized by the element out of cache at any given moment.

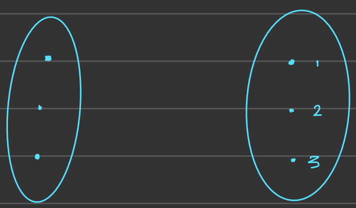
$$\Pr(\text{A request causes a cache miss}) = 1/K$$

$$E(\#\text{ of cache misses per round}) = 1/K \; E(\text{length of round}) \geq 1/K \,(K H_K)$$

Check calc

→ Requests simulate random walk on a complete graph

⇒ Online bi-partite matching

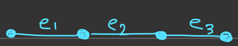


given at start

revealed one v at a time

reveal neighbors as well

→ Maximal matching : A matching which can't be extended to a larger matching



$\{e_2\}$ → maximal

→ $\forall$ |maximal matching| $\geq \frac{1}{2}$ |maximum matching|

- deterministic Algo A is 2-competitive.

→ For deterministic online algo, we can't beat 2-competitive

Randomized online algo :- $1 - 1/e$ → tight bound

* 20th Sep Wed

* Midsem discussion

Q3.

(a) $E(\#\text{ of collisions}) = \binom{n}{2} Pr(\text{collide})$

$\leq \binom{n}{2} \frac{1}{2}$ → 2-univ

$E(\#\text{ of } k \text{ collisions}) \leq \binom{n}{k} \frac{1}{n^{k-1}}$

$Pr\left(\#\text{ of } k \text{ coll} \geq 2\binom{n}{k}\frac{1}{n^{k-1}}\right) \leq \frac{1}{2}$

$Y_j = \#$ balls in bin j $\quad \binom{Y_j}{k} \leq \#\, k$ collisions

(b) $n^{1/k} \longrightarrow \theta\left(\frac{\ln n}{\ln \ln n}\right)$ What is k?

⇒ Online Matching

L R (online)

find a matching

→ Deterministic greedy → Add if possible
Compute a maximal matching $\geq \frac{1}{2}$ (maximum matching)

⇒ Linear program for matching

For every edge $e \to x_e$
a setting of x_e to 0 or 1

$$\max \sum_{e \in E} x_e$$

$$\text{s.t} \sum_{u \in N(v)} x_{(u,v)} \leq 1 \quad \forall u \in V, \quad x_e \in \{0,1\} \quad \forall e \in E$$

↳ NP hard

→ Any assignment to x_e that satisfies the constraints is called feasible sol^n

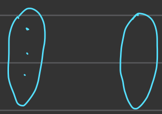
→ For a relaxed version

$$\max \sum_{e \in E} x_e \qquad \text{s.t} \sum x_{(u,v)} \leq 1 \quad \forall u \in V, \quad x_e \geq 0$$

↳ P

↳ OPT turns out to be integral

* 21 Sept Thu

→ 

$M \to$ greedy maximal matching

OPT $\to$ size of maximum matching

$$OPT \leq 2|M|$$

→ Linear Program

x_e edge $e \in G$

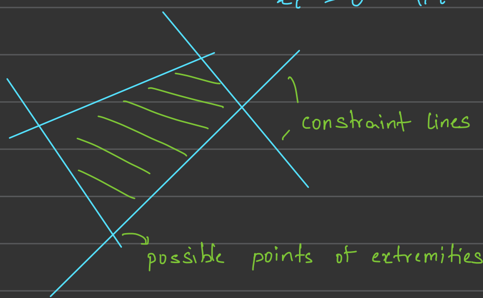
$$\max \sum_{e \in E} x_e \quad , \text{s.t} \quad \sum_{u \in N(v)} x_{(u,v)} \leq 1 \quad \forall v \in V \quad \& \quad x_e \geq 0$$

↳ feasible sol^n if satisfies constraints

$$\rightarrow \max \sum_{i=1}^{n} c_i x_i \qquad \text{s.t} \quad \sum_{j=1}^{n} a_{ij} x_i \leq b_i \quad \forall i \in \{1, \dots m\}$$

$$x_i \geq 0 \quad \forall i \in \{1, \dots n\}$$

↳ Primal max



$$y_1, y_2 \cdots y_m \geq 0$$

$$y_1 \left(\sum_{j=1}^{n} a_{1j} x_j\right) + \dots y_m \left(\sum_{j=1}^{n} a_{mj} x_j\right) \leq \sum_{i=1}^{m} b_i y_i$$

$$= x_1 \left(\sum_{i=1}^{m} a_{i1} y_i\right) + \dots x_n \left(\sum_{i=1}^{m} a_{in} y_i\right) \leq \sum_{i=1}^{m} b_i y_i \quad \rightarrow ①$$

Minimize $\sum_{j=1}^{m} b_j y_j$ s.t $\sum_{j=1}^{m} a_{ji} y_j \geq c_i$ $y_j \geq 0$

$\sum_{i=1}^{n} c_i x_i \leq ① \leq \sum_{j=1}^{m} b_j y_j$ ↳ Dual min

→ Weak duality

OPT(primal) $\leq$ OPT(dual) every feasible dual is an upper bound on the OPT of the primal

→ Strong duality

OPT(primal) $=$ OPT(dual)

→ Construct feasible soln for dual to upper bound primal

⇒ Dual for matching

y_v, $v \in V$

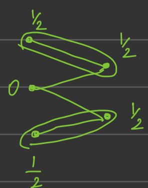
$\min \sum_{v \in V} y_v$ s.t $y_u + y_v \geq 1$ $\forall (u,v) \in E$

$y_u \geq 0$ $\forall u \in V$

→ Given the maximal matching M, construct a feasible sol^n for the dual

- If $(u,v) \in M$ $q_u = 1/2$ & $q_v = 1/2$

$\sum_{u \in V} q_u = |M|$



→ Not satisfied

- Set $y_u = 2q_u$

→ OPT $\leq \sum_{u \in V} y_u \leq 2 \sum_{u \in V} q_u = 2|M|$ $\therefore$ OPT $\leq 2|M|$

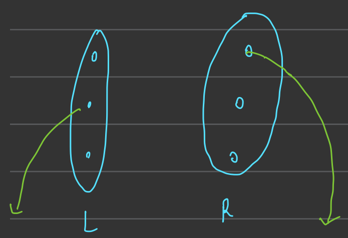
⇒ Generic primal – dual analysis
- Primal LP
- Use algo to construct a dual feasible
- Apply weak - duality

Never actually solve dual or primal LP

⇒ Online Fractional Matching

→ Solve the LP online
→ Let f be the fractional matching
$f \geq (1 - 1/e)\, OPT$

⇒ Waterlevel algorithm

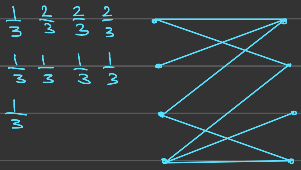


Edges formed can be considered as pipes

Containers which can hold 1 unit of water

Reservoirs containing 1 unit of water

→ Reservoirs fill the container with the lowest quantity or equally divided flow to all containers with least amount at any instant.



Missing 22^{nd} Sat

Fractional matching $(1-1/e)$ OPT

* Mon 25^{th} Sep

⇒ Ranking

→ Outputs a matching M s.t $E(M) \geq (1 - 1/e)\, OPT$

→ Choose a random permutation $\bar{\pi}$ over the vertices of L for each vertex R

match v with $u \in N(v)$ that is unmatched, & has the lowest ranking acc to $\bar{\pi}$

→ Primal

$$\max \sum_{e \in E} x_e$$

$$\text{s.t} \quad \sum x_e \leq 1 \qquad \forall v \in V$$

$$x_e \geq 0 \quad \forall e$$

→ Dual

$$\min \sum_{u \in V} y_u$$

$$\text{s.t} \quad y_u + y_v \geq 1 \qquad \forall (u, v) \in E$$

$$y_u \geq 0 \qquad \forall u \in V$$

→ Weak duality $OPT(\text{primal}) \leq OPT(\text{dual})$

⇒ Alternate form of Ranking

→ for each $u \in L$

Choose $l_u \in [0, 1]$ uar

for each $v \in R$

match v to $u \in N(v)$

that is unmatched & has the lowest l_u

→ for $(u, v) \in E$ $\quad q_u = e^{l_u - 1}, \; q_v = 1 - e^{l_u - 1}$ if $(u, v) \in M$

o/w $\quad q_u = q_v = 0$

→ Obs : $\sum_{u \in L} q_u + \sum_{v \in R} q_v = |M|$

→ If $y_u = \frac{e}{e-1} q_u$ → $E(y_u)$ → feasible for dual

- $E(q_u) \geq$ ___ $u \in L$
- $E(q_v) \geq$ ___ $u \in R$ → $E(q_u + q_v) \geq 1 - 1/e$ This is too be shown

→ $(u,v) \in E$

fix $l_{u'}$ $\forall u' \in L \setminus \{u\}$

Run algo on $G \setminus \{u\}$ Let u^* be the vertex matched to v

l_{u^*} (1 if v is unmatched)

⇒ Claim : If $l_u < l_{u^*}$, then u is matched in G

Proof: ① u is already by the time v arrives

② if ① doesn't happen, u gets matched to v

At v's arrival $\forall u' \in N(v) - \{u\}$ $l_{u'} > l_{u^*} \Rightarrow l_u < l_{u'}$

Conditional

→ If u is matched then $q_u = e^{l_u - 1}$

$E(q_u) \geq \int_0^{l_{u^*}} e^{y-1} dy = e^{l_{u^*}-1} - e^{-1}$

⇒ Claim

$q_v \geq 1 - e^{l_{u^*} - 1}$

→ Run algo on G & $G \setminus \{u\}$ in parallel

The above implies that for $(u,v) \in M$, $l_u < l_{u^*} \Rightarrow q_v = 1 - e^{l_u - 1} \geq 1 - e^{l_{u^*} - 1}$

→ Set of unmatched vertices in G ⊇ Unmatched vertices in $G \setminus \{u\}$

$E(q_u) \geq e^{l_{u^*}-1} - e^{-1} \; \forall u \in L \rightarrow$

$E(q_v) \geq 1 - e^{l_{u^*}-1} \; \forall u \in R \rightarrow$ $E(q_u + q_v) \geq 1 - 1/e$

$E(y_u + y_v) \geq 1 \qquad y_u = \frac{e}{e-1} q_u$

$\therefore \; OPT \leq \sum_{u \in L} E(y_u) + \sum_{v \in R} E(y_v) = \frac{e}{e-1} \sum_{u \in L \cup R} E(q_u) = \frac{e}{e-1} E(|M|)$

* Wed 27th Sep

⇒ Multiplicative Weight Update

→ make Yes/No decisions

→ wrong decisions incur unit cost

→ Minimize the error

→ n experts $e_1, e_2 \dots e_n$ Compare number of mistakes with best expert.

→ m_i^T → total loss over T time steps for expert e_i

→ Minimize $\left[m^T - \min_{i \in [n]} m_i^T \right]$ } → regret

↳ Mistakes made by best expert.

⇒ Omniscient expert - who never makes a mistake

- Once we find the omniscient expert after T time steps, regret stops growing with time

→ We can find expert with $\theta(n)$ regret

→ Take the majority answer

if the answer is correct, then no cost

else

you half the # of potential omniscients, regret ++

regret $\leq \log n$ $\quad m^T \leq m_i^T + \log n$

⇒ Weighted majority algorithm

→ w_i^t → weight assigned to expert i at t^{th} time step

$w_i^{(0)} = 1 \quad \forall i \in \{1, 2 \cdots n\}$ No omniscience

→ if $\sum_{i, e_i \text{ says Yes}} w_i^t \geq \sum_{i, e_i \text{ says NO}} w_i^t$ answer yes

else answer no

→ All experts with wrong answer has $w_i^{t+1} = (1-\epsilon) w_i^t$

⇒ Thm: Let m_i^T be loss incurred by expert i, m^T is loss incurred by the algo

$$m^T \leq 2(1+\epsilon) m_i^T + \frac{2 \ln n}{t}$$ ①

⇒ Thm: Every det algo will make at least twice the # of mistakes as the best expert

→ 2 experts e_1, e_2

Y N → Always Adversary always forces wrong answer at every step

Algo → t errors either e_1 or $e_2 \leq \frac{T}{2}$

⇒ Proof for ①

Potential function $\phi(t)$ → # of potential experts $= \sum_{i=1}^{n} w_i^t$

$\phi(0) = n$

$$\phi(t+1) = \sum_{i, e_i \text{ says } Y} w_i^{(t+1)} + \sum_{i, e_i \text{ says } N} w_i^{(t+1)}$$

$$= (1-\epsilon) \sum_{Y} w_i^t + \sum_{N} w_i^t \quad , \quad \sum_{Y} w_i^t \geq \sum_{N} w_i^t$$

$$\geq \frac{\phi(t)}{2}$$

$$\leq (1 - \epsilon/2)\, \phi(t)$$

→ $w_i^t = (1-\epsilon)^{m_i^t}$ $\qquad (1-\epsilon)^{m_i^T} \leq \phi(t) \leq (1-\epsilon/2)^{m^T} \cdot n$

For randomized algo, we include probability

Wed 4th Oct

⇒ Multiplicative weight sum game

→ Collection of experts $e_1 \cdots e_n$

→ At every time step, make a decision incur unit cost for wrong decision

m^T → Mistakes incurred by algo over T time steps

m_i^T → " " by ith expert

$m^T - \min m_i^T$ = regret

→ Prior det algo → Weighted majority algo

⇒ Theorem : $\forall i \in \{1, 2 \dots n\}$

$$m^T \leq 2(1+\epsilon) m_i^T + \frac{2 \ln n}{\epsilon}$$

⇒ Randomized ver.

→ Cost incurred $m^t \in [-1, 1]$

p^t dist over experts

Choose e_i acc p^t and give the same answer as e_i

$m_1^T \dots m_n^T$ → Cost incurred by the experts

Update $p^t \to p^{t+1}$

$p^t \to \langle p_1^t, \dots p_n^t \rangle$

$\overline{m}^t \to \langle m_1^t, \dots m_n^t \rangle$

→ Expected cost for a round $= \sum_1^n m_i^t p_i^t$
in the 't' time step

→ Expected total cost $= \sum_{t=1}^T \overline{m}^t \cdot p^t$

→ Regret $= \sum_{t=1}^T \overline{m}^t \cdot p^t - \min \sum_{t=1}^T m_i^t$

Algo

→ $\eta \leq 1/2$ $\quad w_i^{(1)} = 1 \quad \forall i \in \{1, 2, \dots n\}$

→ For each t

$$\phi(t) = \sum_1^n w_i(t)$$

$$p^t = \left\{ \frac{w_1^t}{\phi(t)} \dots \frac{w_n^t}{\phi(t)} \right\}$$

→ observe $m_1^t \dots m_i^t$

$\rightarrow w_i^{t+1} = w_i^t (1 - \eta m_i^t)$

$\Rightarrow$ Theorem : $\forall i \in \{1, \dots n\}$

$$\sum_1^T \bar{m}^t p^t \leq \sum_{t=1}^T m_i^t + \eta \sum_1^T |m_i^t| + \frac{\ln n}{\eta}$$

$\rightarrow$ If m_i^t corresponds to gains

$w_i^{t+1} = w_i^t (1 + \eta m_i^t)$

$$\sum_1^T \bar{m}^t p^t \geq \sum_1^T m_i^t - \eta \sum_1^T |m_i^t| - \frac{\ln n}{\eta}$$

$\Rightarrow$ Learning a linear classifier

$\{(\bar{a}_i, l_i)\}_{1 \leq i \leq m}$ $\qquad \bar{a}_i \in \mathbb{R}^n \qquad l_i \in \{\pm 1\}$

Find $\bar{x} \in \mathbb{R}^n$

$\text{sgn}(\sum a_{ij} \cdot x_j) = l_i$

① $|a_{ij}| \leq 1$ $\qquad$ ② $\sum_1^n x_i = 1$ $\qquad$ $x_j \geq 0 \ \forall j \in [n]$

$a_{ij} x_j \geq 0$

* Oct 5th Thursday

⇒ Linear Classifier

→ Labelled samples : $(\bar{a}_1, l_1)$ $(\bar{a}_2, l_2)$ ... $(\bar{a}_m, l_m)$

$\bar{a}_i \in \mathbb{R}^n$

Check once

→ We need a hyperplane seperating the samples

$\sum_1^n w_i x_i \mapsto \forall \bar{a}_i \ \ sgn(\sum a_{ij} w_j) = l_i$

If above has a soln below has one too

→ Vector $w \in \mathbb{R}^n$ $\forall j \in \{1, 2 .. m\}$ $\sum a_{ij} w_j \geq 0$, $|w| = 1$

$w_j \geq 0 \ \forall j$

Assumption

$\exists w^*$ st $\sum_1^n a_{ij} w_j^* \geq \epsilon \quad \forall i \in [m]$

⇒ Algorithm :

$w_i^{(1)} = 1/n \quad \forall i \in [n]$

for $t = 1$ to T → normalize

check if $\bar{a}_j \bar{w}^{(t)} \geq 0 \ \forall j$

if $\exists j$ $\bar{a}_j w^{(t)} < 0$ $\quad w_i^{(t+1)} = w_i^{(t)} (1 + \frac{\epsilon}{2} a_{ji})$

⇒ MWU algorithm

$w_i^{(t)} = 1$

for $t = 1$ to T

$\phi(t) = \sum w_i^{(t)}$

$p^{(t)} = (.. \frac{w_i^{(t)}}{\phi(t)} ...)$

Sample i acc $p^{(t)}$

Observer $m_1^{(t)} ... m_n^{(t)}$

Update $w_i^{(t+1)} = w_i^{(t)} (1 + \eta m_i^{(t)})$

$\Rightarrow$ Theorem :

$$\sum_{1}^{T} m^{(t)} \cdot p^{(t)} \geq \sum_{t=1}^{T} m_i^{(t)} - \eta \sum_{1}^{T} |m_i^{(t)}| - \frac{\ln n}{\eta}$$

$$m_i^{t} = a_{ji} \quad \text{for} \quad \overline{a_j} \cdot w^{(t)} < 0$$

Corollary : for any distribution $p = w^*$

$$\sum_{1}^{T} m^{(t)} \cdot p^{(t)} \geq \sum_{1}^{T} (m^{(t)} - \eta |m^{(t)}|) \cdot p - \frac{\ln n}{\eta}$$

$$0 \geq \sum_{1}^{T} \frac{\epsilon}{2} - \frac{2\ln n}{\epsilon} \Rightarrow \frac{T\epsilon}{2} \leq \frac{2\ln n}{\epsilon} \qquad T \leq \frac{4\ln n}{\epsilon^2}$$

(using assumption above) $\qquad$ # of iterations

End of Online Algo

Markov Chains

$\Rightarrow$ Satisfiability

$\rightarrow \emptyset = C_1 \wedge C_2 \cdots \wedge \underset{\mapsto n \text{ vars}}{C_m}$

$C_1 = l_{i_1} \vee l_{i_2} \vee l_{i_3} \quad \cdots \quad l_{i_k}$

$\rightarrow \exists$ assignment $\bar{x} \in \{0,1\}^n$

$\emptyset(\bar{x}) = 1 \qquad \emptyset = (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee \bar{x}_3)$

↳ Self reducibility

Assign x_i as 0 or 1 and check for result

$\rightarrow$ 2-SAT $\in P$

3-SAT $\in$ NP-complete

$\Rightarrow$ 2 SAT $\quad (\bar{x}_1 \vee \bar{x}_2) \equiv (x_1 \Rightarrow x_2) \equiv (\bar{x}_2 \Rightarrow \bar{x}_1)$

$G(V, E)$

$V = \{x_i, \bar{x}_i \mid i \in [n]\}$

if $l_i \vee l_j \quad (\bar{l}_i, l_j), (\bar{l}_j, l_i) \in E$

If $\exists$ a strongly connected components which has x_i & $\bar{x}_i$ are on the graph, then 2-SAT is not solvable

* Fri 6^{th} Oct

⇒ 2 SAT randomized algorithm.

→ fix $\bar{x} = 0^n$

if $\phi(\bar{x}) = 1$, then done

else choose C_i s.t $C_i(\bar{x}) = 0$

$C_i = l_{i_1} \vee l_{i_2}$

flip l_{i_1} or l_{i_2} uar.

→ $\overset{C_1}{(x_1 \vee x_2)} \wedge \overset{C_2}{(\bar{x}_2 \vee x_3)} = \phi$

(i) All zero

$C_1 = 0 \quad C_2 = 1$

(ii) flip x_2

$C_1 = 1 \quad C_2 = 0$

(iii) flip x_2

$C_1 = 0 \quad C_2 = 1$ → Possible infinite loop

We need to define progress of algo.

⇒ Markov chain

$\{X_t\}_{t \in \mathbb{N}}$, X_t = state of the system at time step t.

Ω = Set of all possible states of the system

↳ finite (in the course)

→ $\{X_t\}_{t \in \mathbb{N}}$ is a Markov chain if

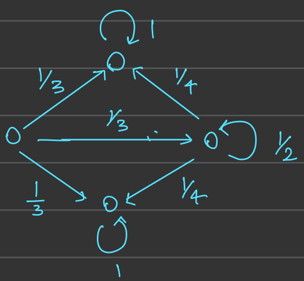
- $\Pr(X_t = a_t \mid X_{t-1} = a_{t-1} \; \ldots \; X_0 = a_0)$
  $= \Pr(X_t = a_t \mid X_{t-1} = a_{t-1})$

  Doesn't mean X_t, X_{t-2} are independant

- $\Pr(X_t = a_t \mid X_{t-1} = a_{t-1})$
  $= \Pr(X_1 = a_t \mid X_0 = a_{t-1})$

→ Directed graph $G(V, E)$ $V = \Omega$

$P_{a_{t-1} a_t}$ → Pr of going to a_t if currently in a_{t-1}



$$P = \begin{bmatrix} & & \\ & P_{ij} & \\ & & \end{bmatrix}$$

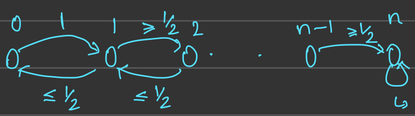
↳ Pr of jumping to i from j in one step

$$\begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 1 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Analysis of random walks

→ States $n+1$

i = assignments that are at a Hamming distance of $n-i$ from a satisfying assgn



↳ satisfying assignment

→ Each move, we can go forward, backward on fixing 1 satisfying assignment as a worse case assignment

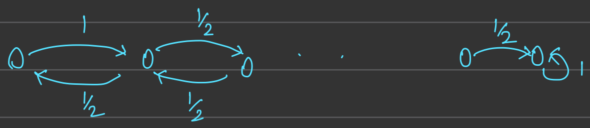
→ At each node, we pick a clause.

Atleast one literal is not the same as satisfying assignment.

$\therefore$ $Pr(\text{forward}) \geq 1/2$ $Pr(\text{backward}) \leq 1/2$

Given the conditions, expected time we reach node n

→ To convert to markov chain, convert all inq. to ½



⇒ M

→ Given markov chain M, E(# of steps to go from 0 to n)

Y_i → time for M to go from i to n

$$E(Y_i) = 1 + \frac{E(Y_{i-1}) + E(Y_{i+1})}{2}$$

$E(Y_0) = 1 + E(Y_1)$ $\qquad E(Y_i) = n^2 - i^2$

$E(Y_n) = 0$ $\qquad$ By running $2n^2$ steps, we find assignment w.p $\geq ½$

Mon 9th Oct

⇒ 3 - SAT

→ $\phi = C_1 \wedge C_2 \cdots \wedge C_m$ $\qquad C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$

$\overset{?}{\exists}\, \bar{x} \in \{0,1\}^n$ s.t $\phi(\bar{x}) = 1$

→ Trivial algorithm $O(2^n)$ $\qquad$ NP complete.

⇒ Deterministic $O(1.732^n)$

→ $x = 0^n$ $\quad\quad C_i(\bar{x}) = 0$

flip l_{i_1} $\quad$ l_{i_2} $\quad$ l_{i_3}

⋮ n step $\Rightarrow 3^n \times$ too bad

→ Obs: Hamming distance from every assignment $\bar{x} \in \{0,1\}^n$ is $\leq n/2$ from 0^n or 1^n

If $n/2$ steps over 0^n not satisfied, start with 1^n

⇒ Randomized 3SAT

→ Choose $\bar{x} \in_r \{0,1\}^n$

If $\phi(\bar{x}) \neq 1$

Let $C_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$ s.t $C_i(\bar{x}) = 0$

n steps $\quad$ Choose $l_{i_1}, l_{i_2}, l_{i_3}$ uar & flip

$$\Pr(\text{success}) \geq \sum_{k=0}^{n} \binom{n}{k} \frac{1}{2^n} \frac{1}{3^k}$$

→ Prob of moving forward until soln

→ Chosen Hamming distance k from correct $\bar{x}$

$$\geq \frac{1}{2^n}\left(\frac{4}{3}\right)^n = \left(2/3\right)^n$$

Repeating $(3/2)^n$ times gives a sol^n w.h.p

$O(1.5^n)$

⇒ Schöning's Algo

→ Suppose $\bar{x}$ has hamming k

→ Prob that algo makes l moves in the wrong direction

$$= \binom{k+2l}{l} \left(\frac{2}{3}\right)^{l} \left(\frac{1}{3}\right)^{k+l}$$

→ Pr we reach assignment $\geq \max_{l} \binom{k+2l}{l} \left(\frac{2}{3}\right)^{l} \left(\frac{1}{3}\right)^{k+l}$

$$\geq \binom{3k}{k} \left(\frac{2}{3}\right)^{k} \left(\frac{1}{3}\right)^{2k} \geq \frac{c}{\sqrt{k}} \frac{1}{2^k}$$

$$\sqrt{2\pi k} \left(\frac{k}{e}\right)^{k} \leq k! \leq 2\sqrt{2\pi k} \left(\frac{k}{e}\right)^{k}$$

→ $Pr(\text{success}) \geq \sum_{k=0}^{n} \binom{n}{k} \frac{1}{2^n} \cdot \frac{c}{\sqrt{k}} \frac{1}{2^k} \geq \left(3/4\right)^{n}$

Running time $O(1.33^n)$

⇒ More about Markov chains

→ Vector π → dbn over Ω

πP → dbn over Ω after 1 time step

πP^n → " n timesteps

P^n : n-step transition matrix

→ If $\exists n$ $P_{i,j}^{n} > 0$, $\exists$ a way to reach j to i in n steps

→ Def^n : A markov chain is irreducible if

$\forall i, j \ \exists n \ P_{i,j}^{n} > 0$

= The diagraph is strongly connected

→ Our satistiability MCs are not irreducible as there is no way out from state n

?

$\Delta > 1$

→ Defⁿ : State j in a MC M is periodic if $\exists\ \Delta > 0$ s.t
$Pr(X_{s+t} = j \mid X_s = j) = 0$ unless $\Delta \mid t$
A MC M is aperiodic if it does not have any periodic states

→ Given $G(V, E)$ – undirected
- G is connected, MC is irreducible
- G is bipartite, MC is periodic

✱ Fri 13th Oct

→ Irreducibility : Directed graph is strongly connected
→ State j is periodic if $\exists\ \Delta > 1$ s.t
$Pr(X_{s+t} = j \mid X_s = j) = 0$ unless Δ / t
- MC is aperiodic if it does not have any periodic states

→ Random walk is aperiodic on an undirected graph iff graph is not bipartite.

⇒ Stationary distribution
A distribution π on the markov chain is stationary if $\pi P = \pi$

⇒ Fundamental thm of Markov chains

Let M be a finite, irreducible, aperiodic MC; then the following hold.
(i) $\exists$ a unique stationary distribution π
(ii) $\lim_{t\to\infty} P^t_{i,j}$ exists and is independent of i
(iii) $\lim_{t\to\infty} P^t_{i,j} = \pi_j = 1/h_{j,j}$
↳ expected time for MC to start and return to j

→ Undirected, connected and non-bipartite graph $G(V, E)$

We need to find $\pi P = \pi$

where $(u,v) \in E$

→ $P_{u,v} = 1/\deg(u)$ $\quad u, v \in E$

$\frac{\deg(u)}{2m}$ $\quad \frac{1}{\deg(u)}$

$\pi(u) = \frac{\deg(u)}{2m}$ $\quad m = |E|$ $\quad \pi P(v) = \sum_{u \in V} \pi(u) P_{u,v}$

⇒ Randomized s-t connectivity

→ Perform a random walk for l steps from s and check if t was visited

→ $O(\log n)$ space RL complexity

→ $O(\log n)$ space deterministically — 2004 Complexity soln

→ $l = O(mn)$

Cover time — Expected time for a random walk to visit all the vertices.

Depends on smallest cut of the graph.

⇒ Claim $\forall (u,v) \in E$

$\underbrace{h_{u,v} + h_{v,u}}_{\text{Commute time}} \leq 2|E|$

M over directed edges $(u,v) \rightarrow \left.\begin{array}{l} u \rightarrow v \\ v \rightarrow u \end{array}\right\}$ states of M'

# of states of M' is $2|E|$ Add self loop to

M' is aperiodic & irreducible. make non bipartite

$P_{u-v, v-w} = \frac{1}{\deg(v)}$ Stationary distribution over $M' \rightarrow \frac{1}{2|E|}$

CNF & DNF

* Mon 16th Oct

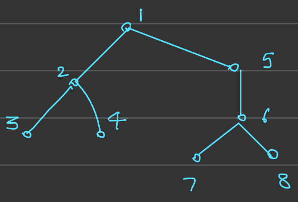
⇒ Randomized Reachability

→ $G(V, E)$ → non bipartite
Starting from s perform on ℓ length random walk & check if t was ever visited.

→ Lemma : $\forall (u,v) \in h_{u,v} + h_{v,u} \leq 2m$

→ Lemma: Cover time of a random walk $\leq 2m(n-1)$
↳ expected time to cover all the vertices in the comp of s

→ Spanning Tree of G : T



DFS traversal → Cover

1 2 3 2 4 2 1 5 6 7 6 8 6 5 1

Random walk expected time of $(h_{uv} + h_{vu})$ for $\forall (u,v) \in T$

$\leq 2m * (n-1) \leq 2n^3$

$\therefore \ell = \delta n^3$, we can check reachability

* Monte Carlo methods Counting & sampling

→ Let V be a quantity that you want to estimate.
X is a (ϵ, δ) approx of V if $\Pr(|X - V| \geq \epsilon V) \leq \delta$

⇒ Fully Poly time Randomized Approx Scheme. (FPRAS)

→ Given x, calculate $V(x)$
An Algorithm A is an FPRAS if it takes x, ϵ, δ as input s.t
$A(x, \epsilon, \delta)$ is an (ϵ, δ) approx of $V(x)$ & it runs in $\mathrm{poly}(\frac{1}{\epsilon}, \ln(1/\delta), |x|)$

⇒ DNF Counting

$\varphi = T_1 \vee T_2 \cdots T_m$ → DNF

$T_1 = l_1 \wedge \cdots l_r$ n literals

→ DNF-SAT $\in P$
→ #DNF $= |\{\bar{a} \mid \varphi(\bar{a}) = 1\}| \in$ #P-complete
(sharp)

→ Trivial : sample assignment in φ, probability of a satisfying assignment v low.

→ Sets $S_1, S_2 \cdots S_m$
Given x, check if $x \in S_i$; Also can sample from S_i
Computation $|\bigcup_1^m S_i|$ $S_i = \{\bar{a} \mid T_i(\bar{x}) = \}$

$\to C_\varphi = \{(i, \bar{a}) \mid T_i(\bar{a}) = 1\}$

$$|C_\varphi| = \sum_{1}^{m} |S_i|$$

<u>Obs</u>: $|C_\varphi| \leq m \left|\bigcup_{1}^{m} S_i\right|$

$S_\varphi = \{(i, \bar{a}) \mid T_i(\bar{a}) = 1,\ \forall j < i\ T_j(\bar{a}) = 0\}$ Obs: $|S_\varphi| = |\bigcup S_i|$

↳ First term satisfied by $\bar{a}$

→ We sample from C_φ and check

Sample i wp $|S_i| / \sum |S_i|$

Sample $\bar{a}$ uar from S_i

$\therefore\ \forall (i, \bar{a})\quad Pr = \dfrac{1}{\sum |S_i|}$

Monte carlo check for $(i, \bar{a}) \in S_\varphi$ $\quad \dfrac{|S_\varphi|}{|C_\varphi|} \geq \dfrac{1}{m}$

Use chernoff bound on k iterations and get number

* Wed 25^{th} Oct

* Monomer - Dimer problem

→ Graph $G(V, E)$ $|V| = 2n$

Monomer occupies 1 vertex

Dimer occupy 2 adjacent vertices

→ Configuration of monomes & dimers s.t every vertex occupies exactly 1 molecule.

→ M_k → Set of configurations with k dimers (2n − 2k monomers)

m_k → # of configs with k dimers

$|M|$ → # of dimers

M_k → k-sized matchings ; M_n → Perfect matchings

⇒ Partition function

$$Z(\lambda) = \sum_M w(M) \qquad w(M) = \lambda^{|M|}$$

$$= \sum_0^n m_k \lambda^k$$

LORF :

Imp for thermodynamics

$$Z(\lambda) = \frac{Z(\lambda_r)}{Z(\lambda_{r-1})} \cdots \frac{Z(\lambda_1)}{Z(\lambda_0)} Z(\lambda_0)$$

$$0 = \lambda_0 > \lambda_1 \cdots > \lambda_r = \lambda$$

→ Gibbs distribution

$$\pi_\lambda(M) = \frac{\lambda^{|M|}}{Z(\lambda)}$$

$$\lambda_i = \left(1 + \frac{1}{n}\right)^{i-1} \lambda_1 \qquad \lambda_1 = 1/|E|$$

→ Estimating $\frac{Z(\lambda_i)}{Z(\lambda_{i-1})}$

→ Sample M acc Gibbs distribution
↳ using MC whose stationary dist is the Gibbs dist

estimate $\left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|}$

$$E(x) = \sum_M \left(\frac{\lambda_{i-1}}{\lambda_i}\right)^{|M|} \frac{\lambda_i^{|M|}}{Z(\lambda_i)} = \frac{1}{Z(\lambda_i)} \sum_M (\lambda_{i-1})^{|M|} = \frac{Z(\lambda_{i-1})}{Z(\lambda_i)}$$

⇒ Metropolis algorithm.

→ Given irreducible, aperiodic MC with uniform stationary dist & given dist π

↓

new MC with the same state space & whose stationary dist is π.

⇒ MC for matchings

M_i — current matching $\forall$ edge $e = \{u, v\}$

- wp $1/2$ $M_{i+1} = M_i$
- wp $1/2$ perform

$M_{i+1} = M_i - \{e\}$ if $e \in M_i$

$M_{i+1} = M_i \cup \{e\}$ if u, v are unmatched

$M_{i+1} = M_i \cup \{e\} - \{e'\}$ if exactly 1 of u, v is matched with e'

$M_{i+1} = M_i$ o/w

wp $\left\{1, \frac{\pi(M_{i+1})}{\pi(M_i)}\right\}$ move to M_{i+1}

∴ for every edge there is one self loop & one outgoing edge

→ Metropolis

Ω $\qquad N(x)$ = set of nbrs of x

$M \geq \max_x |N(x)|$

$$P_{x,y} = \begin{cases} 1/M & x \neq y \quad y \in N(x) \\ 0 & \text{if } y \notin N(x) \\ 1 - \sum\limits_{y \in N(x)} P_{x,y} & \text{if } y = x \end{cases} \implies \frac{1}{M} \min\left\{1, \frac{\pi_y}{\pi_x}\right\} \quad \text{if } y \in N(x)$$

Choose to go $x \to y$ wp.

Our case only moves $-1, 0, 1$

$$\frac{\pi_\lambda(M)}{\pi_\lambda(M')} = \lambda, 1, 1/\lambda \quad \text{only.}$$