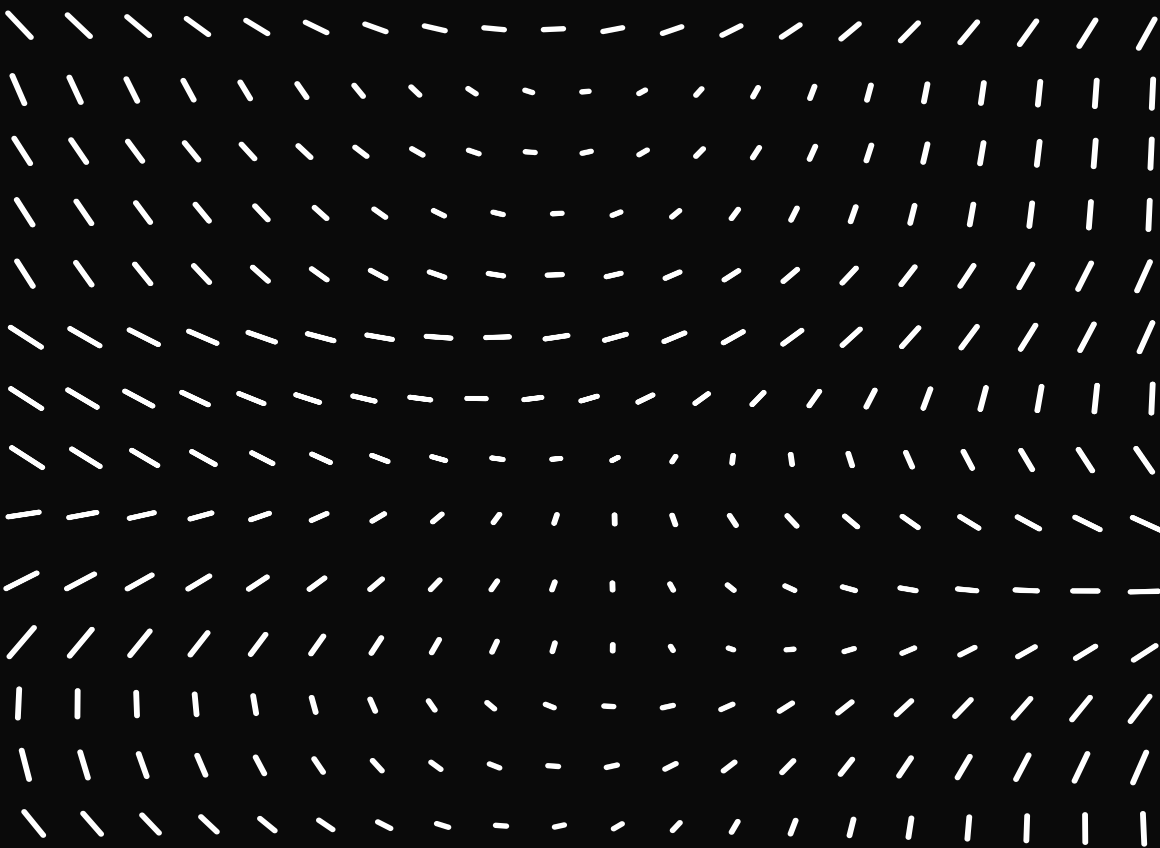


Advanced Algorithm

— Suryajith Chillara

Lecture #1

Assigments 20%

Quiz 1 10%

Mid sem 20%

Quiz 2 10%

Endsem 30%

Viva 10%

Book : Probability and computing Mitzenmacher and Upfal

Verify Matrix Multiplication

Given A, B and C all (n×n mat)

we need to check if $AB \stackrel{?}{=} C_{n \times n}$

→ Multiply A and B and check if the elements are equal complexity : n^{ω} → mat. mul exp.

↳ By Vassi, Almans - Williams

So far, only deterministic algorithm

↳ "fixed" Run time → worst case

↳ No Error (always correct)

↓

we want to relax this and make Bounded run time for most of the time.

Randomised Routing

Randomised algorithm : Primality testing, Rabin - Miller

Trend : Hard Problem → Randomization → Approximation.

$5n^2$ [Bläser] → lower Bound on matrix multiplication.

Updated Question: Is there a rand. algorithm that runs in time $O(n^\delta)$ where $\delta < \omega$. (we want a mostly correct)

If $AB = C$, then $\forall$ vectors v, $ABv = Cv$

else $\exists u$ s.t. $AB \cdot u \neq C \cdot u$ | $u \in$ Nullspace of $AB - C$

$$AB\bar{x} = C\bar{x}$$

$k \times l$, $l \times s$ → kls complexity.

$$\underbrace{AB\bar{x}}_{O(n^2)} \text{ and so } C\bar{x} \implies \begin{bmatrix} L_1(\bar{x}) \\ \vdots \\ L_n(\bar{x}) \end{bmatrix} = \begin{bmatrix} L_1'(\bar{x}) \\ L_2'(\bar{x}) \\ \vdots \\ L_n'(\bar{x}) \end{bmatrix}$$

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \implies \begin{bmatrix} L_1(\bar{x}) - L_1'(\bar{x}) \\ \vdots \\ L_n(\bar{x}) - L_n'(\bar{x}) \end{bmatrix}$$

If $AB = C$ then $\forall i \in [1, n]$, $L_i(\bar{x}) - L_i'(\bar{x}) = 0$ for every substitution of $\bar{x}$ from S^n

If not, $\exists$ a subset for $\bar{x}$ s.t. not all $L_i(\bar{x}) - L_i'(x)$ are zero at that point.

$$\Pr_{\bar{a} \sim S^n}\left[\bigwedge_{i=1}^{n} \left(L_i(\bar{a}) - L_i'(\bar{a}) = 0 \right) \right] \leq \max_{i \in [n]} \Pr\left[L_i(\bar{a}) - L_i'(\bar{a}) = 0 \right]$$

when $\bar{a} \in$ Nullspace of $AB - C$

Bad event: $\forall i,\ L_i(\bar{a}) = L_i'(\bar{a})$

Good Event: $\exists i \in [n]$ s.t. $L_i(\bar{a}) \neq L_i'(a)$

Prob. of $\bar{a}$ being in null space.

$$\Pr_{\bar{a} \in_R S^n}\left[\sum_{j=1}^{n} \alpha_j \cdot a_j = 0 \right] = \frac{|S|^{n-1}}{|S|^n} = \frac{1}{|S|}, \text{ fix } n-1, \ n^{th} \text{ is fixed}$$

$$\leq \frac{1}{|S|}$$

Algo:

1. Pick $\bar{a}$ uniformly random (u.a.r.) from S^n
2. compute $AB\bar{a}$ and $C\bar{a}$
3. Check if pointwise equal
4. If yes, return $AB = C$ $\rightarrow$ 100% correct
   else return $AB \neq C$ $\rightarrow$ correct with prob. $\geq 1 - \frac{1}{|S|}$

Monte Carlo algorithms: correctness not fixed, time fixed

Las Vegas: correctness 1, time unbounded.

Problem: (Polynomial Identity Testing)

Given

$$\left. \begin{array}{l} f(x_1, x_2, \ldots x_n) \\ g(x_1, x_2, \ldots, x_n) \end{array} \right\} \text{ of degree } d.$$

$$x_1^{e_1} x_2^{e_2} \ldots x_n^{e_n}$$

$$\sum_{i=1}^{n} e_i \leq d \qquad \text{no. of coeff. } \binom{n+d}{d}$$

If $f(a) = g(a)$ for $d+1$ points then $f(x) = g(x)$

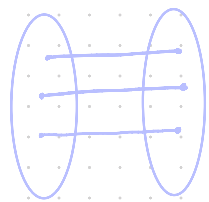
If $f(x) \neq g(x)$ then $\exists$ b in $d+1$ distinct evals

$f(b) \neq g(b)$

Lecture #2

Graph Matching - Bipartite

- Matching M is a subset of edges s.t. for every vertex $v \in V(G)$, $|I(v) \cap M| \leq$ where $I(v)$ is the set of edges incident on v.
- Perfect Matching : $\forall v,\ |I(v) \cap M| = 1$



$V = L \cup R$

$E \subseteq L \times R$

Edmond's Criterion

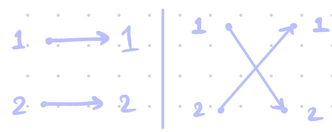
$$(M_{G_{n\times n}})_{ij} = \begin{cases} x_{ij} & \text{if } (i,j) \in E \\ & \quad i \in L,\ j \in R \\ 0 & \text{otherwise} \end{cases}$$

Theorem:

A given bipartite graph has perfect matching (P.M.) if and only if $Det(M_G) \not\equiv 0$

Determinant

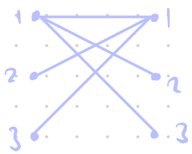
$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \longrightarrow x_{11}x_{22} - x_{21}x_{12}$$



Homework : Prove this

$$Det(X) = \sum_{\sigma \in S_n} (-1)^{sgn(\sigma)} \prod_{i=1}^{n} x_{i\sigma(i)}$$

$$\# inv(\sigma) = \left| \{(i,j) \mid i < j \quad \sigma(i) > \sigma(j)\} \right|$$



$$\longrightarrow \begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & 0 & 0 \\ x_{31} & 0 & 0 \end{bmatrix} \end{array} \equiv 0 \text{ everywhere}$$

$\therefore$ no perfect matching

PM $\Rightarrow$ $Det(M_G) \not\equiv 0$ | $Det(M_G) \not\equiv 0 \Rightarrow$ PM

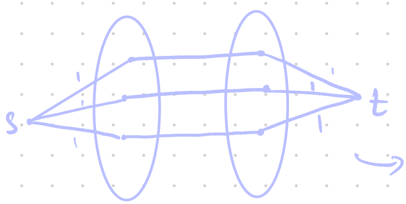
observation: Perfect Matching is infact a permutation.

Further $\prod x_{i\sigma(i)}$ survive iff all $(i, \sigma(i)) \in E$.

Sufficient: To find a point where $Det(M_G) = 0$

σ^*: $\forall i$, set $x_{i\sigma^*(i)} = 1$

Existence criteria, not how to find not discussed yet.



complexity: $O(|C| \cdot (m+n))$

$\hookrightarrow$ max flow $\Rightarrow$ Perfect Matching

Problem: Sequential

Want a parallel algorithm.

Matching in RNC^2

claim: For all $e \in E$: check if $G/\{e\}$ has a PM using EC.

$G \leftarrow G/\{e\}$ if yes

$M \leftarrow M \cup \{e\}$ if No.

Different Ordering will give different perfect Matching

Non-zeroness of a polynomial

Let $f(x_1, x_2, \ldots, x_n)$ be a non-zero polynomial of degree d.

Let $S \subseteq R$ of size $> d+1$

Let $(a_1, a_2, \ldots, a_n)$ be chosen s.t. each a_i is picked and uniformly at Random.

$$\Pr_{\bar{a} \sim S^n}[f(a_1, a_2, \ldots, a_n)] \leq \frac{d}{|S|}$$

\# Zeros$(f) \leq d \cdot |S|^{n-1}$ over S^n.

\# Total points $= |S|^n$

Can be proved using induction. ← Homework

Algorithm for testing $Det(M_G) \not\equiv 0$.

1. Pick a set S of size $100n$
2. Sample $\bar{a}$ independently and uniformly at random from $S^{|E|}$
3. Check if $Det(M_G(\bar{a})) \stackrel{?}{=} 0$
4. If zero ⇒ report $Det(M_G) \equiv 0$

   Else report $Det(M_G) \not\equiv 0$

If $Det(M_G) \equiv 0$ then algorithm makes no error

If $Det(M_G) \not\equiv 0$, then the probability of it be being correct is $1 - \frac{1}{100}$

Randomness $= n \cdot \log|S|$

complexity $= O(|E| \log(100n))$

food for thought for next class:

Having $O(\log(n))$ Randomness = Having no randomness at all.

→ Area called pseudorandomness

Here n is the number of random bits sampled.

Lecture #3

Graph Matching (Bipartite)

There is a random algorithm that finds a perfect matching in $O(\log^2 n)$ parallel time using $O(m \cdot T_{Det}^{\log^2 n}(n))$ processors.

Outline: Assign wts to edges randomly and ensure and extract unique min. wt. matching

Isolation Lemma: Let S be a finite subset of $\mathbb{R}$.

Let $T_1, \dots, T_K \subseteq [m]$ ($[m] = \{1, 2, \dots, m\}$) be all distinct sets

For each element $i \in [m]$, let wt fn assign a wt ind. and U.A.R. from S. wt of a set T_i is given by

$$\sum_{e \in T_i} wt(e)$$

$$Pr(|\text{min wt set}| = 1) \geq 1 - \frac{m}{|S|}$$

Proof: Let E_i be the event that minimum is obtained by two sets U and V and $i \in U$ but not V.

obs.: If min. wt. is not unique then $\exists\, i$, s.t. E_i is true

$$\min_{T_j : i \in T_j} \{wt(T_j)\} = \min_{T_{j'} : i \in T_{j'}} \{wt(T_{j'})\}$$

Bad event: $[\exists i \text{ st } E_i]$

Good event:

$$\Pr\left(\bigcap_{i=1}^{m} \bar{E_i}\right] = 1 - \Pr[\cup E_i] \geq 1 - m \cdot \max_i \{ \Pr[E_i] \}$$

$$\Pr(E_i) \leq \frac{1}{|S|}$$

$$\therefore \Pr\left(\bigcap_{i=1}^{m} \overline{E_i}\right) \geq 1 - \frac{m}{|S|}$$

1. For each each edge $e \in E$, assign a random weight ind. U.A.R. from S.
2. Define matrix W s.t.

$$W_{ij} = \begin{cases} 2^{wt((i,j))} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise} \end{cases}$$

Lemma: Let M_0 be the unique min wt PM in G.

Let $r = wt(M_0)$. Then

$$(i,j) \in M_0 \quad \text{iff} \quad \frac{\operatorname{Det}(W^{(i,j)})}{2^r} \cdot 2^{w(i,j)} \text{ is odd.}$$

3. For each edge $(i,j) \in E$, compute $\frac{\operatorname{Det}(W^{(i,j)})}{2^r} \cdot 2^{wt((i,j))}$

(in parallel)

$$\operatorname{Det}(W) = \sum_{\substack{\sigma \in S_n \\ \sigma \in PM(G)}} (-1)^{sgn(\sigma)} 2^{\sum_{i=1}^{n} wt(i, \sigma(i))}$$

$M_0, M_1, \ldots, M_K$

$$\mathrm{Det}(W) = \sum_{\sigma \in S_n} (-1)^{\mathrm{sgn}(\sigma)} \prod_{i=1}^{n} W_{i\sigma(i)}$$

$$\mathrm{Det}(W) = \mathrm{sgn}(M_0) \cdot 2^{wt(M_0)} + \mathrm{sgn}(M_1) \cdot 2^{wt(M_1)} + \dots + \mathrm{sgn}(M_k)\, 2^{wt(M_k)}$$

$$\forall i \neq wt(M_i)$$

Lecture #4

$$W_{ij} = \begin{cases} 2^{w(i,j)} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

$wt : E \longrightarrow S$

u.a.r. + Ind.

Conditioned on

1. PM existence check (Edmond's criteria)
2. Unique min wt guarantee (Isolation lemma)

Claim : $\dfrac{\text{Det}(W)}{2^{wt(M_0)}}$ is odd

Lemma : $M_0 \leftarrow$ min wt PM , $r \leftarrow wt(M_0)$

$(i,j) \in M_0$ if and only if $\text{Det}\left(\dfrac{W^{(i,j)}}{2^r}\right.$

Lecture #5

Randomised Routing (Parallel)

Hypercube : $G = (\{0,1\}^n , \{ (u, u \oplus e_i) \mid u \in \{0,1\}^n, i \in [n] \})$

$\oplus$ → xoR

e_i → $(0, 0, \ldots, \underset{i^{th}}{1}, \ldots, 0)$ → elementary vector

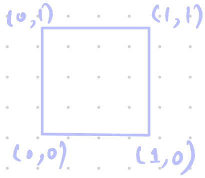
$\underline{n = 2}$

$\{0, 0\}$

$\{0, 1\}$

$\{1, 0\}$

$\{1, 1\}$



We want to route packets on this graph from one vertex to other

→ Permutation Routing (oblivious/independent Routing)

Bit fixing scheme

source	destination
011011	110010

MSB to LSB

$\underline{0}11011$

↓

$11\underline{1}011$ → $11001\underline{1}$ → 110010

source   dest.   LSB to MSB

$$\begin{array}{ccc} 0110\underline{11} & & 110010 \\ \downarrow & & \uparrow \\ 01\underline{1}010 & \longrightarrow & \underline{0}10010 \end{array}$$

Task: Route all packets from every source to every dest as early as possible

There could be waiting at the nodes

Every node can hold unlimited packets and each edge carriers atmost one packet.

Lemma: There are permutations $\Pi \; \{0,1\}^n \to \{0,1\}^n$ for which bit fixing scheme takes $\frac{2^{n/2}}{n}$ time steps to route all the packets.

Possible: Hamming distance n

$$(a_1, a_2, \dots, a_n) \longrightarrow (\bar{a}_1, \dots, \bar{a}_n)$$

$$(x_{n/2}, 0_{n/2}) \longrightarrow (0_{n/2}, x_{n/2}) \;\; \forall \;\; x_{n/2} \in \{0,1\}^{n/2} \Big\} \begin{array}{l} 2^{n/2} \text{ many} \\ \text{out of } 2^n \end{array}$$

Claim: Each of these packets need to go through $(0_{n/2}, 0_{n/2})$

At each time step atmost n packets can be routed from any node

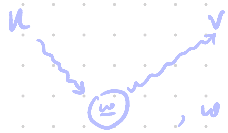
Time to route all packets $(x_{n/2}, 0_{n/2}) \to (0_{n/2}, x_{n/2})$ is at least $\frac{2^{n/2}-1}{n}$.

Theorem: For every deterministic protocol, $\exists$ a permutation that needs at least $\sqrt{\frac{2^n}{n}}$ time.

Randomization: For every permutation, all packets can be routed in at most $O(n)$ time steps with high probability.

Lecture # 6

Theorem: There is a randomised protocol that routes all packets to their dest. in at most $15n$ times w.h.p.



, $w \in path(u \to v)$

Claim: $\exists\, a \in [n]$ s.t.

$u_i = w_i \quad \forall\, i > a$

$v_i = w_i \quad \forall\, i \leq a$

src $= u_1 u_2 \dots u_n$

$v_1 u_2 u_3 \dots u_n$

$v_1 v_2 u_3 \dots u_n$

$\vdots$

$v_1 v_2 \dots v_{n-1} u_n \longrightarrow$ dest: $v_1 v_2 \dots v_n$

$Path(v) =$ Path from $v \to \pi(v)$ using BFS

Lemma 1: For $v \neq v'$, $Path(v)$ and $Path(v')$ do not meet again once they diverge.

Lemma 2: Let S be the set of vertices w s.t. $Path(u)$ and $Path(w)$ intersect. Then P_v takes at most $n + |S|$ time steps to reach $\pi(v)$

Lemma 3: # of w s.t. Path(w) intersects with path $v \to \sigma(v)$ is at most $6n$ w.p. $\geq 1 - 2^{-n}$

similarly, $\sigma(v) \longrightarrow \pi(v)$ wp $\geq 1-2^{-n}$ intersections $\leq 6n$

$$\begin{tikzcd}[row sep=small] & y_v = x \oplus e_a \\ x \arrow[ur] \arrow[dr] & \\ & y_w = x \oplus e_b \end{tikzcd}$$

Lecture # 7

Chernoff bound:

For some $t \in \mathbb{R}_{\geq 0}$,

$$\Pr[X > (1+\delta)\mu] = \Pr[Xt > (1+\delta)\mu t]$$
$$= \Pr[e^{Xt} > e^{(1+\delta)\mu t}]$$
$$\leq \frac{E(e^{Xt})}{e^{(1+\delta)\mu t}}$$

$$E(e^{Xt}) = E[e^{t\sum_{i=1}^{n} X_i}] = \prod_{i=1}^{n} E[e^{tX_i}]$$

$$E[e^{tX_i}] = e^{t\cdot 1} \cdot p_i + (1-p_i)\cdot e^{t\cdot 0}$$
$$= p_i(e^t - 1) + 1$$

$$\prod_{i=1}^{n} p_i(e^t-1)+1 \leq \prod_{i=1}^{n} e^{p_i(e^t-1)} = e^{\sum p_i(e^t-1)} = e^{\mu(e^t-1)}$$

$e^t = 1+\delta$

$$\therefore E[X > (1+\delta)\mu] \leq \left[\frac{e^{(e^t-1)}}{e^{(1+\delta)t}}\right]^{\mu}$$

minimize / derivative to get rid of t

$$= \left[e^{(e^t-1)-(1+\delta)t}\right]^{\mu}$$

set $e^t = 1+\delta$

$$\geq \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{\mu}$$

Home work : Chernoff bound for error reductions

$$x_i = \begin{cases} 1 & 1/2 \text{ prob.} \\ -1 & 1/2 \text{ prob.} \end{cases}$$

$$X = \sum x_i$$

$$Pr[X > R]$$

T_1 T_2

PIT

$P_{R_1} \stackrel{?}{=} P_{R_2} \rightarrow$

T_1 T_2

x_0 x_0

h

$h+1$

$h+1$

h

$h+1$

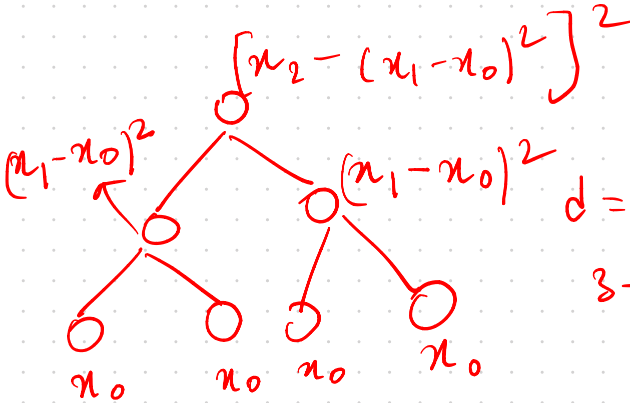
$$\prod_{i=1}^{k}\left(x_{h+1} - P_{v_i}\right) \qquad \prod_{i=1}^{k}\left(x_{h+1} - P'_{v_i}\right)$$

$h+1$ - vars

$x_0 - x_h$

$$\prod_{i=1}^{k}\left(x_{h+1} - c_i\right) \qquad \prod_{i=1}^{k}\left(x_{h+1} - c_i\right)$$

$$P\left[P_{R_1} - P_{R_2} = 0\right] \le \frac{d}{|S|}$$



$d = 3$

3-variate polynomial

$$x_2^2 - 2x_2(x_1 - x_0)^2 + (x_1 - x_0)^4$$

$$x_2^2 - 2x_2(x_1^2 - 2x_1x_0 + x_0^2) + x_1^4 - 4x_1^3x_0 + 6x_1^2x_0^2 - 4x_1x_0^3 + x_0^4$$

$$x_2^2 - 2x_1^2x_2 + 4x_0x_1x_2 - 2x_0^2x_2 + x_1^2 - 4x_0x_1^3 + 6x_0^2x_1^2 - 4x_0^3x_1 + x_0^4$$

Lecture #8

Satisfiability

Is there an assignment to $x_1, \dots, x_n$ from $\{T, F\}^n$ s.t. the formulae evaluates to true. (Decision Problem)

Its called SAT

SAT is NP-complete (Cook-Levin Theorem)

MaxSAT

Conjuntive Normal Forms: AND of ORs

↳ $c_1 \wedge c_2 \wedge c_3 \dots c_m$

each c_i is or of literals

$c_i = (x_{i_1} \vee x_{i_2} \dots x_{i_K})$

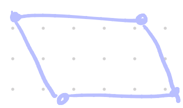
MaxSAT: Given a CNF, what is the max no. of clauses that can be satisfied

↳ (optimization Problem)

" solving MaxSAT solving SAT"

↳ "MaxSAT is at least as hard as SAT"

$$z_j = \begin{cases} 1 & \text{if clause } c_j \text{ is sat} \\ 0 & \text{o/w} \end{cases}$$



$$y_j = \begin{cases} 1 & \text{if } x_i \leftarrow T \\ 0 & \end{cases}$$

$x_i \in \{T, F\}$ u.a.r, independently

Assume each clause has exactly k literals

$$(\quad) \wedge (\quad) \wedge (\quad)$$

$$m\left(1 - \frac{1}{2^k}\right) \geq \frac{m}{2}$$

$$\Pr\left(\sum z_j < \frac{m}{2}\right) \leq$$

Approximation algorithms

Vertex Cover: subset of V s.t. each edge gets covered

Independent set: subset of vertex s.t. no two vertex have an edge between them.

performance ratio: $\inf_{I} \frac{\max_A(I)}{\max_{opt}(I)}$ / $\min_{I} \frac{m_A(I)}{m_{opt}(I)}$

Random Assignment gives a $\frac{1}{2}$-approx. algorithm for SAT w.p. ε

→ Good event happens w/ non-zero probability.

Lecture # 8

K-CNF $\quad C_1 \wedge C_2 \cdots \wedge C_m$

$C_i = x_{i_1} \vee x_{i_2} \cdots \vee x_{i_K}$

$x_1, \cdots, x_n \in \{T, F\}$ $\qquad \frac{3m}{4} \qquad \frac{m}{4}$

$E(\#\text{ clauses sat}) =$

$(x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2)$ $\qquad \frac{2m}{3} \quad \frac{2m}{3} \quad \frac{m}{2} \quad \frac{m}{2}$

$\longrightarrow$ Method of conditional expectation $\qquad \frac{m}{4} \quad \frac{3m}{4}$

$x_1 \quad x_2 \cdots x_{10}$

$m \geq m_* \geq m/2$

can find an assignment that sat $\geq \frac{m_*}{2}$ clauses

$\Rightarrow$ half approx algorithm

$\frac{2^{n-k}}{2^n}$ $\qquad n$

$(C_{11} \vee C_{12} \vee \cdots \vee C_{1K}) \wedge (C_{21} \vee C_{22} \vee \cdots \vee C_{2K}) \wedge (C_{31} \vee C_{32} \vee \cdots \vee C_{3K}) \cdots (C_{m1} \vee C_{m2} \vee \cdots C_{mK})$

with $\left(1 - \frac{1}{2^K}\right)$ written above each of the first three clauses.

$C_{ij} \in \{x_1, \cdots, x_n\}$

$m\left(1 - \frac{1}{2^K}\right)$

what is C_j^+ & C_j^- ?

$C_1 \wedge C_2 \wedge \cdots \wedge C_m$

$\mathbb{E}[Z_1] = \Pr[Z_1 = 1]$

$= 1 - \frac{1}{2^k} \geq \frac{1}{2}.$

Randomised Rounding

$$y_i = \begin{cases} 1 & \text{w.p. } \hat{y}_i \\ 0 & \text{o/w} \end{cases}$$

$$z_j = \begin{cases} 1 & \text{w.p. } \hat{z}_j \\ 0 & \text{o/w} \end{cases}$$

$$\mathbb{E}[\# \text{ clauses sat} \mid x_1 = T]$$

$$= \mathbb{E}\Big[\sum_j z_j \,\Big|\, x_1 = T\Big]$$

$$= \sum_j \mathbb{E}[z_j \mid x_1 = T]$$

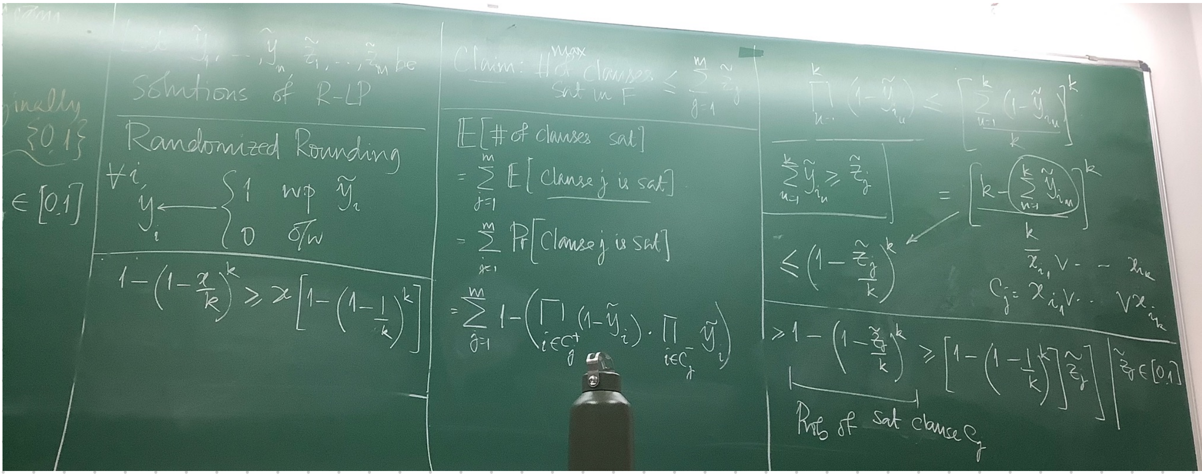
$$= \sum_j \Pr[z_j = 1 \mid x_1 = T]$$

$$=$$

$$\mathbb{E}\Big[\sum_{j=1}^{m} z_j\Big] = \sum \mathbb{E}[z_j] = \sum_{j=1}^{m} \Big(1 - \prod_{i \in C_j^-} \hat{y}_i \prod_{i \in C_j^+} (1 - \hat{y}_i)\Big)$$

Lecture #9

$$E\left[\sum_{j=1}^{m} z_j\right] = \sum E[z_j] = \sum_{j=1}^{m}\left(1 - \prod_{i\in C_j^-} \hat{y}_i \prod_{i \in C_j^+}(1-\hat{y}_i)\right)$$



$$1-\left(1-\frac{\hat{z}_j}{K}\right)^K \geq \left[1-\left(1-\frac{1}{K}\right)^K\right]\hat{z}_j \quad \Big| \quad z_j \in [0,1]$$

$$E(\#\text{ clauses sat}) \geq (1-e^{-1})\sum_{j=1}^{m}\hat{z}_j$$

Lecture #10

Garey Johnson

Maximum Cut

Find S^* such that S^* maximizes $\max_{S \subseteq V} |E(S, \bar{S})|$

Problem is NP-hard.

Attempt-1:

For each vertex v put v in S w.p. $1/2$

$$X_i = \begin{cases} 1 & \text{if } e_i \in E(S, \bar{S}) \\ 0 & \end{cases} \quad , e_1, e_2, \ldots, e$$

$$E[|cut(S, \bar{S})|] = \sum_{i=1}^{m} E(X_i)$$

$$= \sum_{i=1}^{m} \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2}$$

$$= \sum_{i=1}^{m} \frac{1}{2} = \frac{m}{2}$$

Lecture #11

$$\max \sum_{(i,j)\in E} \left(\frac{x_i - x_j}{2}\right)^2 \quad \text{or } 1 - x_i x_j$$

subject to

$$\forall i, \; x_i \in \{-1, 1\}$$

Lecture #12

Min cut - Karger's algorithm

Min - Cut from Max flow algorithm : $\leq n^2 O(\underbrace{m}_{|C|}(m+n))$

claim: G has at least $Kn/2$ edges if $|\text{min cut}| = K$

sub claim : Every cut has a degree at least K.

↳ if not, cut size < K.

C: What is the probability of obtaining this specific cut? (C is specified by K edges)

$E_i \rightarrow$ Event that edges from C are not picked in the i^{th} step , $1 \leq i \leq n-2$

$$Pr\left[\bigcap_{i=1}^{n-2} E_i\right] = 1 - Pr\left[\bigcup_{i=1}^{n} \bar{E}_i\right]$$ → union bound may not give good results

$$= Pr[E_1] \cdot Pr[E_2 | E_1] \dots Pr\left[E_{n-2} \,\middle|\, \bigcap_{i=1}^{n-3} E_i\right]$$

$$Pr[E_1] = 1 - \frac{K}{m} \geq 1 - \frac{2K}{Kn} = 1 - \frac{2}{n}$$

super nodes : Set of Nodes

super edges : set of edges

<u>Init</u> : $T \leftarrow$ super nodes, $F \leftarrow$ super edges

$T \leftarrow V$, $F \leftarrow E$

$$\Pr\left[E_j \,\middle|\, \bigcap_{i=1}^{j-1} E_i\right] \geq 1 - \frac{k_j}{\frac{k_j(n-j+1)}{2}}$$

$$\Pr\left[\bigcap_{i=1}^{n} E_i\right] \geq \prod_{j=1}^{n-2} \left(1 - \frac{2}{n-j+1}\right)$$

$$= \prod_{j=1}^{n-2} \left(\frac{n-j-1}{n-j+1}\right) = \frac{2}{n(n-1)} \geq \frac{2}{n^2}$$

Lecture #13

while $|V| > 2$:

1. Pick an edge at random and merge the end-points
2. Repeat 1. (Pick an edge from the edges that have survived)

Return the edges between the last two nodes.

$\mathcal{E}_i$: Event that edges from cut are not merged after i^{th} step

want $\Pr\left[\bigcap_{i=1}^{n-1} \mathcal{E}_i\right]$

<u>claim</u>: If a graph has a cut of size K then $|E| \geq \frac{Kn}{2}$

cut: $e_1, e_2, \dots, e_K$

$$\Pr[\mathcal{E}_1] = 1 - \frac{2}{|E|} \geq 1 - \frac{2}{n}$$

$$\Pr(\mathcal{E}_2 \mid \mathcal{E}_1)$$

$E^{(i)}$: Edges surviving after i^{th} step

$$|E^{(1)}| \geq \frac{K(n-1)}{2}$$

$$|E^{(d)}| \geq \frac{K(n-d)}{2}$$

$$\Pr(\mathcal{E}_d \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cdots \cap \mathcal{E}_{d-1}) = 1 - \frac{K}{|E_{d-1}|}$$

$$Pr(\mathcal{E}_i \mid \mathcal{E}_1 \cap \mathcal{E}_2 \cap \dots \cap \mathcal{E}_{i-1}) \geq 1 - \frac{2}{n-(i-1)}$$

$$Pr\left[\bigcap_{i=1}^{n-1} \mathcal{E}_i\right] = \prod_{j=1}^{n-1}\left(1 - \frac{2}{n-(j-1)}\right) = \frac{2}{n(n-1)}$$

$$\left(1 - \frac{2}{n(n-1)}\right)^{t-1} \frac{2}{n(n-1)}$$

$$\left(1 - \frac{2}{n^2}\right)^{t} \rightarrow \left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2} \frac{2t}{n^2}} \rightarrow e^{-2t/n^2}$$

Algorithm	Runtime	Prob
Karger	$t \cdot n^2$	$1 - e^{-2t/n^2}$
Max flow	$n^2(m+n)\lvert C\rvert$	1

Lecture #14

Counting Problems

- Count the # of Hamiltonian cycles in graphs
- Count the # of solution to a SAT instance
- Count the # of PMs in a given graphs
- Count the # of K-cliques in a graph

$\Pi \in NP^{dec} \longleftrightarrow$ Counting Version

$\#P$ — "Sharp" P

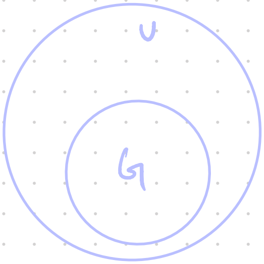
↳ At least as hard as NP

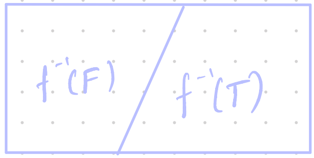
↳ Existence of PTIME algos is questionable

$$Det(X) = \sum_{\sigma \in S_n} (-1)^{sign(\sigma)} \prod_{i=1}^{n} x_{i\sigma(i)}$$

$$Perm(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)}$$

Count the no. of sat instance "approximately" and in polynomial time





$U \to$ Universe

Set of interest G

If $a \in U$ is picked up, computing $f(a)$ is easy

NP : is $|f^{-1}(T)| \geq 1$

#P : $|f^{-1}(T)|$?

$a_1, \ldots, a_N$ u.a.r, ind. from U.

$$Pr(a_i \in G) = \frac{|G|}{|U|}$$

$$y_i = \begin{cases} 1, & a_i \in G \\ 0 & o/w \end{cases}$$

If $z = \frac{|U|}{N} \sum_{i=1}^{N} y_i$, then $E(z) = |G|$

estimator R.V.

$E(z) = |G|$

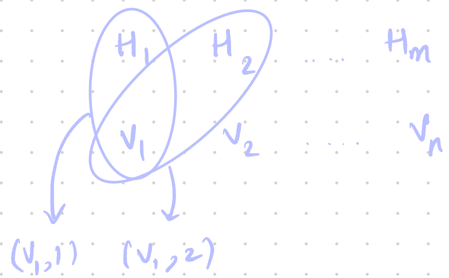
PRAS

(ε, δ) - RAS

DNF → ORs of AND

Lecture #15

DNF $\overset{?}{\longleftrightarrow}$ CNF



Claim : $\dfrac{|\hat{U}|}{\left|\bigcup_{i=1}^{m} H_i\right|} \leq m \rightarrow$ At most m repitions possible.

$$\hat{U} = \biguplus_{i=1} H_i$$

$$G = \{(v, i) \mid f(v, i) = 1\}$$

where $f(u, i) = \begin{cases} 1 & \text{if } i = \min\{j \mid (v, j) \in \hat{U}\} \\ 0 & \text{o/w} \end{cases}$

$$|G| = \left|\bigcup_{i=1}^{m} H_i\right|$$

Uniform Sampling from $\hat{U}$

- First pick $i \in [m]$ at random w.p. $\dfrac{|H_i|}{|\hat{U}|}$

$> \varepsilon$

Lecture #16

Problem : Count the no. of Independent set in a given Graph

NP-complete decision Problem : $\exists \ |I| \geq K$?

$e_1, e_2, \ldots, e_m$ ← Enumeration of edges in arbitrary order

$G_1 = (V, E \setminus \{e_1\})$

$G_2 = (V, E \setminus \{e_2\})$

$\vdots$

$G_m = (V, E \setminus \{e_m\})$

$|I(G_i)|$ → No. Independent sets in Graphs G_i

$$|I(G)| = \frac{|I(G)|}{|I(G_1)|} \times \frac{|I(G_1)|}{|I(G_2)|} \times \ldots \times \frac{|I(G_{m-1})|}{|I(G_m)|} \times \underbrace{|I(G_m)|}_{2^n}$$

$E(G_i) \subset E(G_{i-1})$

$I(G_{i-1}) \subset I(G_i)$ contains independent sets with u, v where $e_i = (u, v)$

$|I(G_i) \setminus I(G_{i-1})| \leq |I(G_{i-1})|$

$$r_i = \frac{|I(G_{i-1})|}{|I(G_i)|}$$

$$I(G_0) = 2^n \prod_{i=1}^{m} r_i$$

$U \leftarrow I(G_i)$

$G \leftarrow I(G_{i-1})$

$X_a = 1$ if the a^{th} sample $\in G$

$$\left| Pr(X_a = 1) - \frac{|I(G_{i-1})|}{|I(G_i)|} \right| \leq \frac{\varepsilon}{6m}$$

Overall, we want (ε, δ)-approximation

For each r_i, if we have $\left(\frac{\varepsilon}{2m}, \frac{\delta}{2m}\right)$-approx., it is sufficient

$$Pr\left(|\tilde{r}_i - r_i| \geq \frac{\varepsilon}{2m} r_i \right) \leq \frac{\delta}{2m}$$

$$1 - \frac{\varepsilon}{2m} \leq \frac{\hat{r}_i}{r_i} \leq 1 + \frac{\varepsilon}{2m}$$

w.p. $\geq 1 - \delta$,

$$\left(1 - \frac{\varepsilon}{2m}\right)^m \leq \prod \frac{\hat{r}_i}{r_i} \leq \left(1 + \frac{\varepsilon}{2m}\right)^m$$

Chapter 10/11 – Mitzenmacher Upfal

Lecture #17

マルコブチアンス

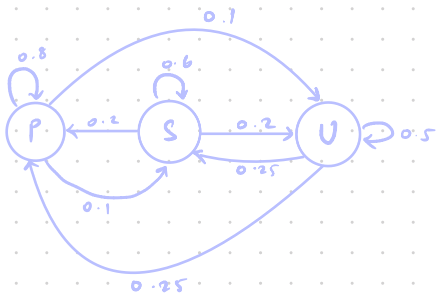
なんでもない

$G=(V,E)$

Lecture # 18

0 1 2 3 4

	0	1	2	3	4
0	0	0	0	0	1
1	0	0	0	1-p	p
2	0	0	1-p	p	0
3	0	1-p	p	0	0
4	1-p	p	0	0	0



	P	S	U
P	0.8	0.1	0.1
S	0.2	0.6	0.2
U	0.25	0.25	0.5

$$\begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.25 & 0.25 & 0.5 \end{bmatrix} \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

$$= \Big[$$

$$0.25 \times 0.8 + 0.25 \times 0.2 + 0.5 \times 0.25$$

$$0.2 + 0.05 + 0.125$$

$$0.375$$

$\pi = \pi P \longrightarrow$ eigenvector with value 1

Stationary distribution

"Limiting" distribution (if it exists)

$$\pi_i = \sum_{K=1}^{n} \pi_K \cdot P_{K,i}$$

$\underbrace{\text{Doubly stochastic matrices}}_{\text{Row \& column sum both are 1}}$ have unique stationary distribution

It may not have a limiting

Lecture #20

$$h_j = \frac{1}{3}(1+h_{j+1}) + \frac{2}{3}(1+h_{j-1})$$

$$2(h_j - h_{j-1}) = (h_{j+1} - h_j) + 3$$

$$h_j = 2^{n+2} - 2^{j+2} - 3(n-j)$$

Lecture #21

Q.
- can I take MFDS?

Markov Chains

- Random walks on undirected graphs

$$P_{v_1 \to v_2} = \begin{cases} 1/d_{v_1} & \text{if } (v_1 \to v_2) \in E \\ 0 & \text{o/w} \end{cases}$$

$$\pi = \pi P$$

$$\pi_i = \sum_{k=1}^{n} \pi_k P_{ki} \quad , \forall i \in [n]$$

$$\pi_i = \sum_{k : v_i \in N(v_k)} \pi_k \cdot \frac{1}{d_k}$$

If the MC is "strongly connected" (irreducible) "finite" and "aperiodic" then we have unique stationary distribution.

$$\pi_i = \frac{1}{h_{ii}}$$

r_{ij}^t = Prob. that $X_t = j$; $X_0 = i$ and $\forall s \in [1, t-1]$ $X_s \neq j$

h_{ij} = Exp. time to first visit state j for the first time starting from state i.

$$h_{ij} = \sum_{t>0} t \cdot r_{ij}^t$$

① ② ...... ⓝ

$$P_{i \to i+1} = \frac{i}{i+1}$$

$$P_{i \to 1} = \frac{1}{i+1}$$

$$r_{11}^{t} = \left(\prod_{i=1}^{t-1} \frac{i}{i+1} \right) \cdot \frac{1}{t+1}$$

$$h_{11}^{t} = \sum_{t>0} \frac{1}{t+1} = \infty$$

$P_{ij}^{t} \neq 1$ → This is not correct acc. to sir, check this once

For a state s:

Period(s)

$= \gcd(t \mid (P^t)_{s,s} > 0\}$

State is aperiodic if period$(s) = 1$

- If $|q| < n$ customers
  w.p. λ, a new customer joins
- If $|q| > 0$, the w.p. μ the head of the queue is served
- w.r.p., q is unchanged

Gambler's ruin:

Player 1 cannot loss more than ℓ_1 dollars

Player 2 cannot loss more than ℓ_2 dollars

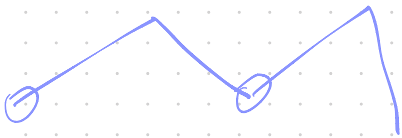
what is the prob. that P_1 gains ℓ_2 dollars before P_2 gains ℓ_1 dollors?

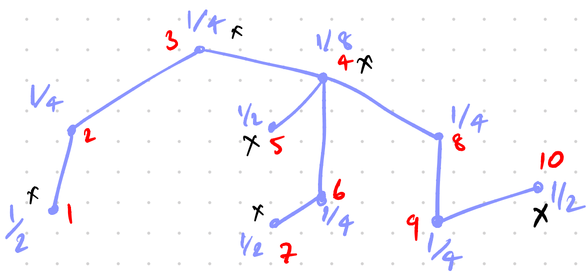
Side question:
Prove a non bipartite graph (SC, finite) is aperiodic.

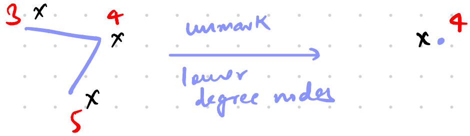
Lecture # 22

Maximal Independent sets (MIS) – luby's Algorithm

Can this be parallized







If v is marked, the include it in S

update $I \leftarrow I \cup S$

update G: $V \leftarrow V \setminus (S \cup N(S))$

Claim: At every stage, at least half of the edges are removed

Lemma: In an iteration, if a vertex w is marked, then w is added to S w.p.

Lemma: Let v be a good vertex. Then prob. that a neighbour of v gets marked is at least $(1 - e^{-1/6})$

A vertex v is good if $|\{w \in N(v) \mid deg(w) > deg(v)\}|$ is at most $\frac{2}{3} d_v$

Good Edge: At least one end point is good.

Lemma: # good Edge $\geq \frac{|E|}{2}$

Lemma 1 + Lemma 2 $\Rightarrow$ $\geq \frac{(1-e^{-1/6})}{2}$ with which a good vertex $\in N(S)$

Lecture #23

(final class)

MIS

1. $I \leftarrow \{\}$
2. While V is not empty:
   2a. For all $v \in V$ (in parallel)
   If $deg(v) = 0$, then $I \leftarrow I \cup \{v\}$, $V \leftarrow V \setminus \{v\}$
   else mark v w.p. $\frac{1}{2deg(v)}$
   2b. For all $(u,v) \in E$ (in pair)
   If both end points are marked,
   2c. For all $v \in V$ (in parallel)
   Add $v \in S$ if v is marked
   2d. $I \leftarrow I \cup S$
3. Return I

"Good" Vertex $\geq \frac{d_v}{3}$ neighbours have deg $\leq d_v$

Book for probabilistic Methods: Alon & Spencer (2nd chapter)

PCP Theorems - NP certificates