22/2/24
Binomial Heap

Delete min $\rightarrow O(\log n)$ — In Fibonacci Heap

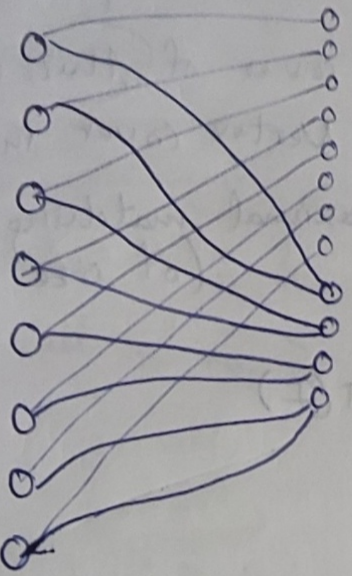
Dec key $\rightarrow O(\log n) \rightarrow O(1)$

Insert $\rightarrow O(1)$

19/3/24 Vertex Cover:-

Bipartite graph:- — Counter example for 1) greedy - higher degree 1st algo.

$l = 8$ vertices



R_i — i

R_1 deg of each vertex $= 1$

R_2 — deg of each vertex $= 2$

OPT $= l$

$$Alg(I) = \sum_{i=1}^{l} \left\lfloor \frac{l}{i} \right\rfloor$$

$$\sum_{i=1}^{l} \left\lfloor \frac{l}{i} \right\rfloor \leq l \sum_{i=1}^{l} \frac{1}{i} \leq l \log l$$

Correctness of algo here means that the algo produces a valid sol^n - Here a vertex cover.

Correct 20/3/24

Algo:-

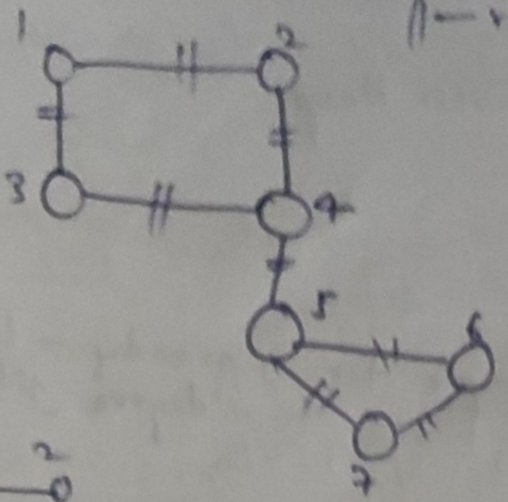
1. $C \leftarrow \phi$
2. while $E \neq \phi$

2.1 Pick any $e = (u, v) \in E$

2.2 $C \leftarrow C \cup \{u, v\}$

2.3 Delete all the edges incident to either u or v

3. Return C

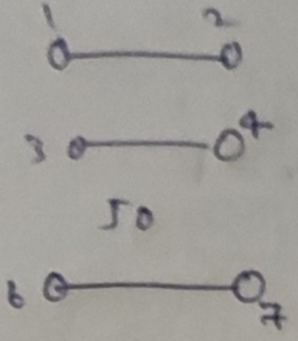


$C = \phi$

$C = \{3, 4\}$

$C = \{1, 2, 3, 4\}$

$C = \{1, 2, 3, 4, 6, 7\}$



size of optimal vertex cover of this (any) matching
$\leq$ size of optimal vertex cover in G.

We have chosen a maximal matching here (of size k)

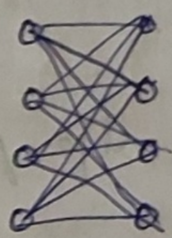
$$OPT(I) \geq k$$

$$Algo(I) = 2k \leq 2 * OPT(I)$$

Tight example:-

Complete Bipartite graph:-

$OPT(I) =$



Integer LP:

1. → Formulate ILP
2. → Relax the variables to make it LP
3. Solve LP
4. Go back to Main (Vertex Cover) (VC) problem

Weighted vertex cover

$$x_v = \begin{cases} 0 & \text{if } v \text{ is not in the vertex cover (VC)} \\ 1 & \end{cases}$$

1. ILP:

$$\min \sum_{v \in V} w_v x_v$$

s.t for each edge $e=(u,v)$

$$x_u + x_v \geq 1$$

$$x_v \in \{0,1\} \ \forall \ v \in V$$

2. $x_v \geq 0 \ \forall \ v \in V$

3. $x^* = (x^*_{v_1}, x^*_{v_2}, \ldots, x^*_{v_n})$

4. $\underbrace{1 \geq x^*_{v_i}}_{\text{bcuz it is the min of LP}} \underbrace{\geq 0}_{\text{LP cond}^n}$

Choose all $v_i \ni x^*_{v_i} \geq \frac{1}{2}$

21/3/24

~~T.S.T~~

Let $S = \{v \mid x^*_v \geq \frac{1}{2}\}$

T.S.T

1) S is a vertex cover of G

2) Time Complexity

3) Approx factor

Pf:

1) As for every $e=(u,v)$, $x^*_u + x^*_v \geq 1$

3) $OPT_{LP}(I) = \sum_{v \in V} w_v x^*_v \geq \sum_{v \in S} w_v x^*_v \geq \sum_{v \in S} w_v \times \frac{1}{2} = \frac{1}{2} w(S)$

$$w(S) \leq 2 OPT_{LP}(I) \leq 2 OPT_{ILP}(I)$$

Set Cover — There is no better algo. than $\log n$

Input:- (X, F)

$F \subseteq P(X)$ ∋ $\bigcup_{C \in F} C = X$

$X = \{x_1, x_2, \ldots, x_n\}$

$F = \{C_1, C_2, \ldots, C_m\}$

Output:- $F' \subseteq F$ ∋ $\bigcup_{C \in F'} C = X$

Pricing technique

Distribute the cost 1 among the new elements ~~that~~ th get added to the set cover.

$$\sum_{i=1}^{n} C_{x_i} = |F| \longrightarrow ①$$

C_x — the cost of $x \in X$

Here we mean the the cost associated to the element x_i

Let the sets be added in the order $S_1, S_2, \ldots, S_k$

Let $x \in S_i$

$$C_x = \begin{cases} \dfrac{1}{\left|S_i - \bigcup_{j=1}^{i-1} S_j\right|} & \text{if } x \in S_i - \bigcup_{j=1}^{i-1} S_j \\ 0 & \text{if } x \in \bigcup_{j=1}^{i-1} S_j \end{cases}$$

$$\sum_{x \in X} C_x \leq \sum_{S \in F^*} \sum_{x \in S} C_x \longrightarrow ②$$

23/3/24

Greedy Set Cover (X, F)

1. $U = X$
2. $C = \phi$
3. While $U \neq \phi$
4. select a $S \in F$ that maximises $|S \cap U|$
5. $U = U \setminus S$
6. $C = C \cup \{S\}$
7. return C

$X = \{x_1, x_2, \ldots, x_{10}\}$

$F = \{ C_1 = \{x_1, x_2, x_3, \ldots, x_8\}, C_2 = \{x_7, x_8, x_9, x_{10}\}, C_3 = \{x_8, x_9, x_{10}\} \}$

Ex: Here, $c_{x_i} = \frac{1}{8}$ $i = 1, 2, \ldots, 8$ $c_{x_9} = \frac{1}{2}$ $c_{x_{10}} = \frac{1}{2}$

$C = \{S_1, S_2, \ldots\}$

$|C| = \sum_{x \in X} c_x$

Claim:- $\sum_{x \in S} c_x \leq H(|S|)$ where $H(d) = 1 + \frac{1}{2} + \cdots + \frac{1}{d}$

~~where $H(d) = 1 + \frac{1}{2} + \cdots + \frac{1}{d}$~~

From ②,

$$|C| \leq \sum_{S \in C^*} \left(\sum_{x \in S} c_x \right) \longrightarrow ③$$

$$\leq \sum_{S \in C^*} H(|S|)$$

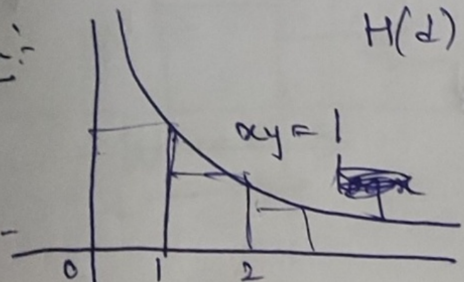
$$\leq \sum_{S \in C^*} H(d)$$ d is the size of the largest set

$$= H(d) \sum_{S \in C^*} 1$$

$$= H(d) \times OPT$$

$$|C| \leq H(d) \times OPT$$

Claim:- $H(d) = \Theta(\log d)$ $\int_1^n \frac{1}{x} \cdot dx$



$1 + \frac{1}{2} + \cdots + \frac{1}{d} < \log n <$

lower sum $< \log n <$ Upper sum

For any set $S \in F$

$u_i = |S \setminus (S_1 \cup S_2 \ldots S_i)|$

$u_0 = |S|$

Let k be the min index such that $u_k = 0$

$\rightarrow u_{i-1} \geq u_i$

$\rightarrow (u_{i-1} - u_i)$ elements of S are covered for 1st time by S_i for $i = 1, 2, \ldots, k$

$$\sum_{x \in S} c_x = \sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{\left| S_i - \bigcup_{j=1}^{i-1} S_j \right|}$$

$$\le \sum_{i=1}^{k} (u_{i-1} - u_i) \times \frac{1}{\left| S - \bigcup_{j=1}^{i-1} S_j \right|}$$

$\left| S_i - \bigcup_{j=1}^{i-1} S_j \right| \geq \left| S - \bigcup_{j=1}^{i-1} S_j \right|$ — S_i covers more new elements than S. That is why it is chosen by the greedy algo.

$$\sum_{x \in S} c_x \le \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{u_{i-1}}$$

$$= \sum_{i=1}^{k} \left(\sum_{j=u_i+1}^{u_{i-1}} 1 \right) \cdot \frac{1}{u_{i-1}} \qquad j \le u_{i-1}$$

$$\le \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{j} = \sum_{i=1}^{k} \left[H(u_{i-1}) - H(u_i) \right]$$

$$= H(u_0) - H(|S|)$$

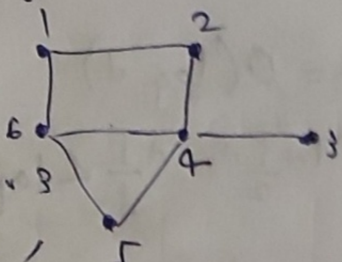
<u>Corollary:-2</u>

<u>Dominating Set</u>

$V' \subseteq V$ is a Dominating set if $V = V' \cup \{\text{neighbours of } V'\}$

$X = \{1, 2, 3, 4, 5, 6\}$

$F = \{ \overset{\text{for 1}}{\{1,2,6\}}, \overset{\text{for 2}}{\{1,2,3,4\}}, \overset{\text{for 3}}{\{2,3,4\}}, \ldots \}$



Here $w: V \to \{0, 1\}$

Roman domination:-

$R: V \to \{0, 1, 2\}$ — the constraint is that every vertex with wt 0 must have a neighbour of wt 2.

3/9/24

MAX SAT

$\phi(x_1, x_2, \ldots, x_m) = C_1 \wedge C_2 \ldots \wedge C_l$

$C_1 \to w_1$, $C_2 \to w_2$, $\ldots$, $C_l \to w_l$

I_1 be the set of clauses which are true for the assignment $x_1 = x_2 = \ldots = x_m = 1$

I_2 " " " $\ldots = 0$

$I_2 \supseteq \bar{I}_1$ $\quad W_1 + W_2 \geq$ sum of wts of all clauses $\geq$ OPT

$\therefore \text{Max}(W_1, W_2) \geq \frac{1}{2} \times \text{OPT}$

4/9/24

Knapsack (Items with wt $\geq W$ are removed)

Items :- $I_1, I_2, \ldots, I_n$ | W

$w_1, w_2, \ldots, w_n$

$v_1, v_2, \ldots, v_n$

a) Counter ex (max value pick)

$w_n = W = v_n$

$w_i = 1 \quad v_i = W - 1$

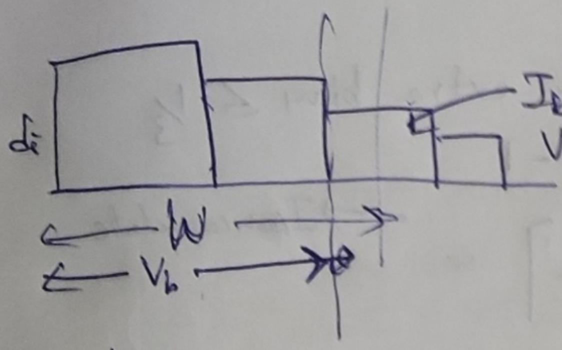
b) Counter ex. $\left(d_i = \frac{v_i}{w_i}\right)$

$w_1 = v_1 = W$

$w_2 = 1, v_2 = 1 + \epsilon$

$v_j \leq v_a \quad \forall j$ picked by a)

$v_i \leq v_a \quad \forall i$ picked by b)



I_k

$V_b + V_a \geq V_b + V_k \geq \text{OPT}$

$\therefore \text{Max}(V_a, V_b) \geq \frac{1}{2} \times \text{OPT}$

Bin packing

$S = (\underbrace{S_1, S_2, \ldots, S_n}_{\text{size}})$

~~Largest sized object~~

→ Arrange the objects in descending order

→ Pick an object and try to fitt it in bin that was filled the earliest.

Sci Pa

$\rightarrow s_1 \geq s_2 \geq \ldots s_i \geq s_n$

s_i is the item that went first into $(OPT+1)^{th}$ bin

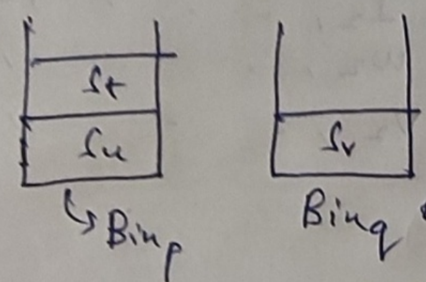
T.S.T $s_i \leq 1/3$

<u>Pf:</u> Assume $s_i > 1/3$

$$\underbrace{1, 2, \ldots, k}_{\text{single item bins}} \quad \underbrace{k+1 \ldots\ldots OPT}_{\text{2 item bins}}$$

T.S.T $0 \leq k \leq OPT$

Suppose otherwise s_i — item in $(OPT+1)^{th}$ bin



$p < q$

Bins filled according to algo

$s_u \geq s_v$

$s_t + s_i > 1$

9/4/24

$\rightarrow$ Size of each item that fits in extra bins $\leq 1/3$

$\rightarrow$ # of items in extra bins $\leq OPT - 1$

Size of our sol^n $\leq OPT + \left\lceil \frac{OPT-1}{3} \right\rceil$ $\longrightarrow$ Incomplete

~~Scheduling~~

→ Scheduling jobs on identical $||^{le}$ machines (Williamson & Schmoys)

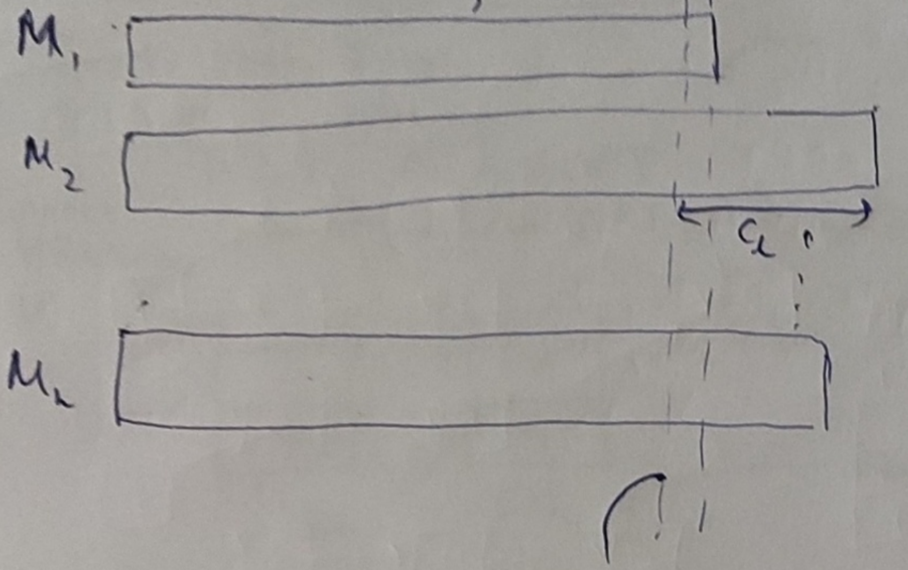
$J_1 \quad J_2 \quad \ldots \quad J_m$

$p_1 \quad p_2 \quad \ldots \quad p_m$ — time to

$M_1 \quad M_2 \quad \ldots \quad M_m$ finish time of job J_i is C_i

Minimize $C = \max_{i=1 \text{ to } n} C_i$

Algo I:- A scheduling such that u can't take the last job of any machine and give it to some other machine & make it better



$$\max(p_1, p_2, \ldots, p_n) \leq OPT$$

$$\text{Algo} \leq OPT + C_l$$
$$\leq 2 * OPT$$

long job –
$$p_x > \frac{1}{km} \sum_{j=1}^{n} p_j \quad \text{remaining are short jobs}$$

no of long jobs $\leq km$ (otherwise $\sum_{\text{long jobs}} p_x > \sum_{j=1}^{n} p_j \Rightarrow \Leftarrow$)

Find optimal for longer jobs – $O(m^{km})$

Use some 2-factor algo on shorter jobs

$$\text{Algo} \leq p_l + \sum_{\substack{j=1 \\ j \neq l}}^{n} \frac{p_j}{m} \leq p_l + \sum_{j=1}^{n} \frac{p_j}{m}$$

$$\leq \sum_{j=1}^{n} \frac{p_j}{mk} + \sum_{j=1}^{n} \frac{p_j}{m} = \left(1 + \frac{1}{k}\right) \sum \frac{p_j}{m} \leq \left(1 + \frac{1}{k}\right) OPT$$
$$= (1+\epsilon)\, OPT$$

10/4/23

Scheduling jobs with deadlines on a single machine (No preempt^n)

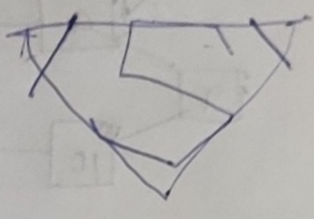
J_1	J_2	.	.	.	J_n
r_1	r_2	-	.	.	r_n
p_1	p_2	.	.	.	p_n
d_1	d_2	-	-	.	d_n

lateness $L_i = C_i - d_i$

Objective :- minimize the max lateness

$$L_{max} = \max_i L_i$$

Algo :- ~~Schedule~~ do do the task with least deadline first



Result :-

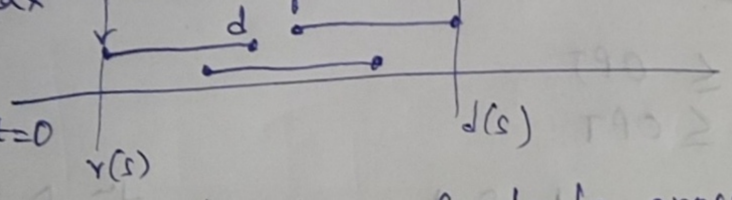
→ Let S be a subset of jobs

→ $r(S) = \min_{j \in S} r_j$

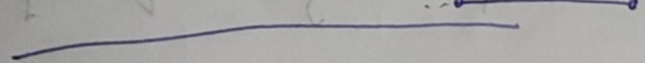
$p(S) = \sum_{j \in S} p_j$

$d(S) = \max_{j \in S} d_j$

Let L^*_{max} → denote opt value $\qquad L^*_{max} \geq r(S) + p(S) - d(S)$



Let j be the last job in S to be processed



$L_j \leq r(S) + p(S) -$

$d_j \leq d(S)$ $\quad C_1$

$r(S) + p(S) \leq C_j$

$C_j - d_j \leq L^*_{max}$

$r(S) + p(S) - d(S) \leq L^*_{max}$

We can assume all $d_j < 0$ (by shifting)

Let S be the set of jobs executed during $[t, C_j]$, C_1

The earliest pt in time the machine was processing without any idle time.

C_j is the job of max lateness

$r(S) = t$

$p(S) = C_j - t$

$= C_j - r(S) \Rightarrow C_j = p(S) + r(S)$

$$L^*_{max} \geq r(S) + p(S) - d(S) \geq r(S) + p(S) \geq c_j \longrightarrow (1)$$

Let $S = \{j\}$

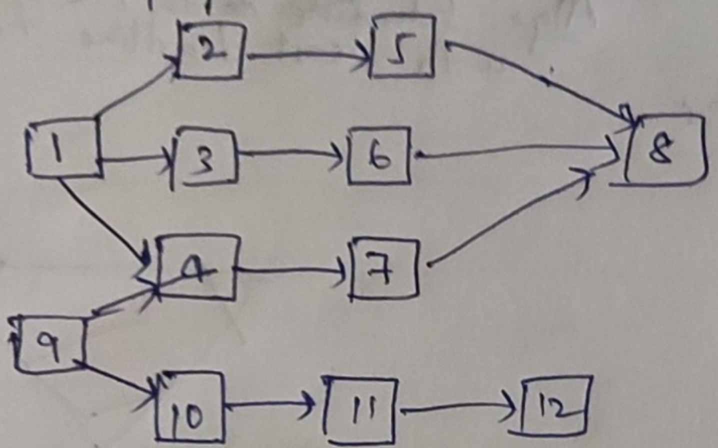
$$L^*_{max} \geq r_j + p_j - d_j$$

$$L^*_{max} \geq -d_j \longrightarrow (2)$$

$$2L^*_{max} \geq c_j - d_j$$

11/4/24

Scheduling problem



→ 1 unit time
→ m machines

Algo:- Pick the available one

1	2	3	4	5	
	2	4	6	8	
1	3	5	7	13	
9	10	11	12		

Claim:-

Approximation factor 2:-

Busy chain ≤ OPT
None busy chain ≤ OPT

TSP — In General, no const factor approximat^n factor possible
└ For Metric TSP :- The one following triangle inequality

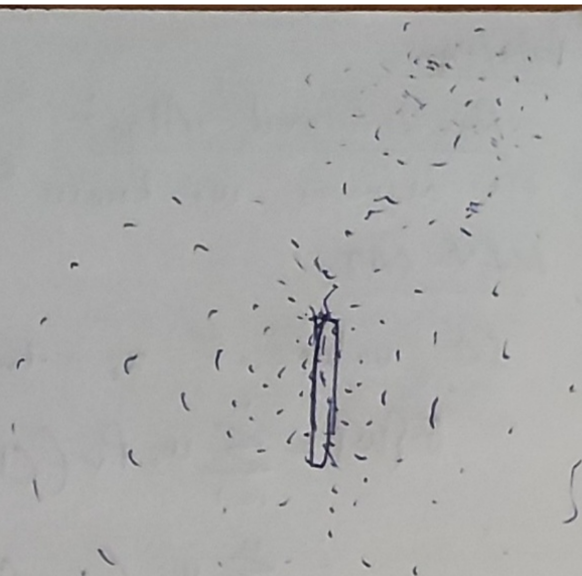
$$C(H^*_G) \geq C(MST)$$

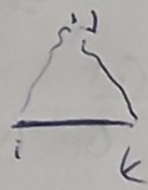
$$C(MST) \leq C(\text{Spanning Tree}) \leq C(H^*_G)$$

TSP - Greedy based on nearest neighbour - 2 factor - read on own

$S \subseteq V$

Claim: $C(H_S^x) \leq C(H_G^x)$





10/4/24

Randomized Algo:-

He assumes we know rand Quicksort Analysis.

MAX-SAT

Set value of x_i = true with probability $1/2$

$$\mathbb{E}[W] = \sum_{i=1}^{n} w_i \Pr(\text{Clause } C_i \text{ is true})$$

l_i is length of C_i — the no of literals

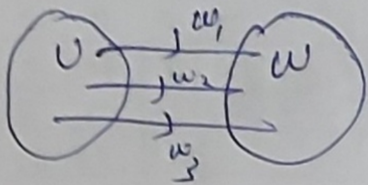
$$= \sum_{i=1}^{n} w_i \left(1 - \left(\tfrac{1}{2}\right)^{l_i}\right)$$

$$\geq \sum_{i=1}^{n} w_i \left(\tfrac{1}{2}\right) \geq \frac{1}{2} \sum_{i=1}^{n} w_i \geq \frac{1}{2} \times OPT$$

Max Cut

$G = (V, E)$

$w : E \to \mathbb{R}^+$ $w(e) = w_{e_i}$



Algo:-

Put $v \in V$ into U or W with probability $1/2$.

$$\mathbb{E}[w] = \sum_{(i,j) \in E} w_{ij} \times (\text{prob of edge } (i,j) \text{ in cut})$$

$$= \sum_{(i,j) \in E} w_{ij} \times \frac{1}{2} \geq \frac{1}{2} \times OPT$$

Derandomizat^n

$$\mathbb{E}[W] = \mathbb{E}[W \mid x_1 \to \text{true}] \underbrace{\Pr(x_1 \to \text{true})}_{1/2} + \mathbb{E}[W \mid x_1 \to \text{false}] \underbrace{\Pr(x_1 \to \text{false})}_{1/2}$$

if $\mathbb{E}[W \mid x_1 \to \text{true}] \geq \mathbb{E}[W \mid x_1 \to \text{false}]$,

$$\leq \mathbb{E}[W \mid x_1 \to \text{true}]$$

Slly

$$\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ldots, x_i \leftarrow b_i] = \mathbb{E}[W \mid x_1 \leftarrow b_1, \ldots, x_{i+1} \leftarrow \text{true}] \times \Pr(x_i \leftarrow \text{true})$$

$$+ \mathbb{E}[\ \ldots, x_{i+1} \leftarrow \text{false}] \times \Pr(x_{i+1} \leftarrow \text{false})$$

$E[w \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ldots, x_i \leftarrow b_i]$ Let the i^{th} i variables be set. Now, we want to set $(i+1)^{th}$ ~~varia~~ variable

$= \sum_{j=1}^{m} w_j \cdot \Pr(j^{th} \text{ clause is satisfied})$

$= \begin{cases} 1 & \text{if } C_j \text{ is satisfied by one of } x_k, 1 \le k < i \text{ already} \\ 1 - \left(\frac{1}{2}\right)^{(\text{no of unset variables in the clause})} & \text{o.w} \end{cases}$

$x_i \leftarrow$ true with probability $p (> \frac{1}{2})$ $\forall i$

$C_j = (\underbrace{\quad\quad}_{\#a \text{ direct variables}} \quad \underbrace{\quad\quad}_{\#b \text{ complement variables}})$

$\Pr(C_j \text{ is satisfied}) = 1 - p^b (1-p)^a$

$\geq 1 - p^{b+a}$

$\geq 1 - p^2$

If $C_q = \bar{x_j}$ with $wt = w_i$ if $w_\ell > w_i$, do nothing

$C_\ell = x_j$ $wt = w_\ell$ else, replace x_j with $\frac{1}{x_j}$.

$E[w] = \sum_{i=1}^{m} w_i \cdot \Pr(i^{th} \text{ clause is satisfied})$ — Analysis assuming there are no s s.t. $C_s = \bar{x_i}$ for any i.

$= \min(p, 1-p^2) \sum w_i \geq \min(p, 1-p^2)\, opt$

If there is an $\bar{x_j}$, the $OPT \leq \sum_{j=1}^{m} w_j - \sum_{i=1}^{n} v_i$

$E[W] = \sum_{j=1}^{m} w_j \Pr[\text{clause } C_j \text{ is satisfied}]$ where $v_i = \begin{cases} \text{weight of clause } \bar{x_i}, \text{ if both clause } x_i \text{ \& clause } \bar{x_i} \text{ exist} \\ 0 \quad \text{o.w} \end{cases}$

$\geq \sum_{j \in U} w_j \cdot \Pr[\text{clause } C_j \text{ is satisfied}]$

$\geq p \sum_{j \in U} w_j = p\left(\sum_{j=1}^{m} w_j - \sum_{i=1}^{n} v_i\right) \geq p \cdot OPT$

23/4/24

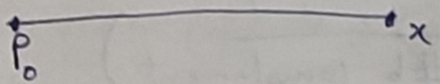
Computat'al Geometry

Convex hull :- A convex area enclosed by a $S \subseteq V$ that encl[oses] V and has min area.

Assume no 3 pts are collinear & no 4 pts are on a circle

Graham's scan Cormen

$O(n) + \underbrace{O(n\log n)}_{\text{sort w.r.t angle}} + O(n)$ an element is pushed / popped only once



Read correctness.

Jarvis march :-

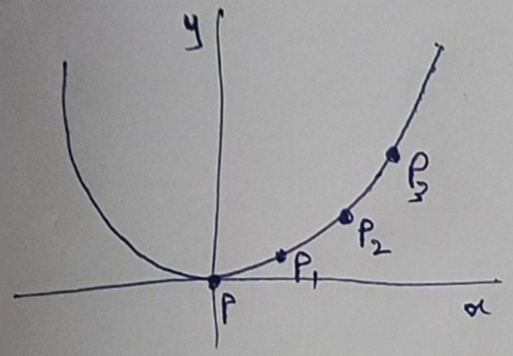
$O(nh)$ — h - no of hull pts

$h = n$ when, n pts lie on the same circle.

Merging 2 convex hull into one :- (H.W)

P.T finding convex hull will take atleast $\Omega(n\log n)$.

Pf: If we can find convex hull of $P=\{(x_1,x_1^2),(x_2,x_2^2), \dots (x_n,x_n^2)\}$, in $o(n\log n)$, then we can sort $[x_1, x_2, \dots, x_n]$ in $o(n\log n)$.



$S=\{P_1, \dots, P_n\}$ is the convex hull of S.

Line Segments

Plane ~~line~~ Sweeping algo:-

Input:- Line segments $\{P_{istart}, P_{iend}\}_{i=1(1)n}$

Output:- Intersecn pts.