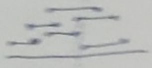
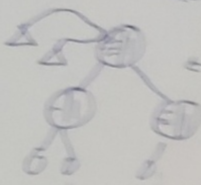
→ Interval Trees

- Set of intervals containing a point



Each node contains
- point of division || (say x)
- 2 AVL trees (L & R)

which contains the left or right side endpoints of intervals intersecting x



← goes down max. tree = $O(\log n)$

(again [illegible] $O(\log n)$)

Add time is [illegible] ??

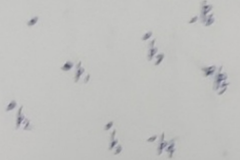
Preprocess time $O(n \log n)$

Query time $O(\log n + K)$

K = no. output / no. K intersecting segments

← then in search, we perform interval query (range search) in AVL tree

- Ham-Sandwich Theorem (in 2-d halfs)

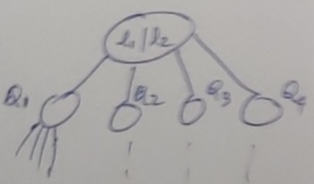


i> Take a horizontal line that divides points half-half

ii> Take another line that divides both sides in half (existence from 2d ham-sandwich theorem)

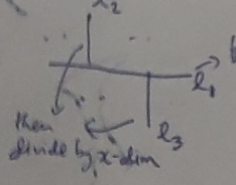
iii> 4 quadrants with equal points

↓ create tree like this

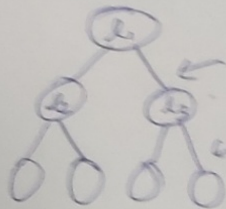


search - $O(n^{\log_4 3})$

- Kd trees (not in syllabus)



→ create tree

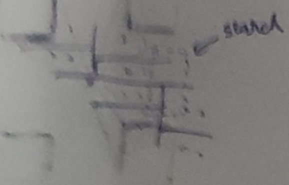


← divide by y coord

← divide by x coord

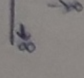
keep alternating

Aim: given a rectangle, find # of points in it.



All points are in half level

In search, after 2 (d) levels, we have structure like



(search space) ← half space search

...rs. params, we use taylor series

28/2/24

Binomial Heap

Delete min → O(log n) — In Fibonacci Heap

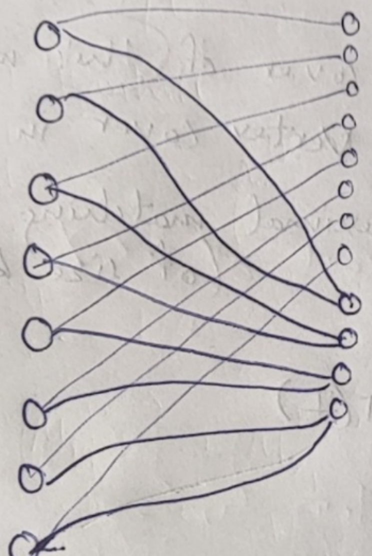
Dec key → O(log n) → O(1)

Insert → O(1)

19/3/24 **Vertex Cover:-**

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

$\lambda = 8$ vertices



R_i

R_1 deg of each vertex = 1

R_2 — deg of each vertex = 2

OPT = l

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

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

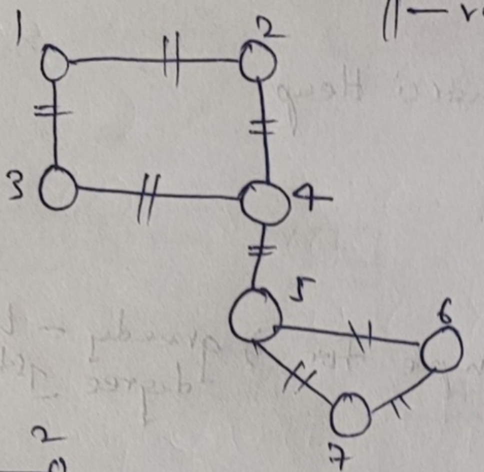
Correctness of algo here means that the algo produces a valid solⁿ — Here a vertex cover.

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

||— removed

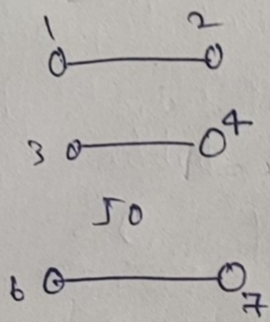


$C = \phi$

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

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

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



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

any

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

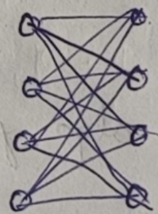
$OPT(I) \geq k$

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

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) problem (VC)

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

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 alg. than $\log n$

Input:- (X, F)

$$F \subseteq P(X) \ni \bigcup_{C \in F} C = X \qquad X = \{x_1, x_2, \ldots, x_n\}$$

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

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

Pricing technique

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

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

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 $x \in S_i$

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

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

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

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

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

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

From (2),

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

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

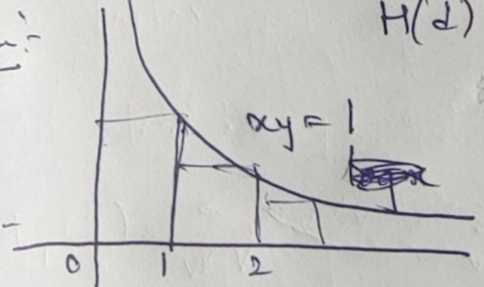
$\leq \sum_{S \in C^*} H(d)$ d is the size of the largest set

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

$= H(d) * OPT$

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

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



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

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 $i = 1, 2, \ldots, k$

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

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

$\left|S_i - \bigcup_{j=1}^{i-1} S_j\right| \ge \left|S - \bigcup_{j=1}^{i-1} S_j\right|$ — S_i covers mo[re] elements that S. That is why it is chosen by greedy algo.

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

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

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

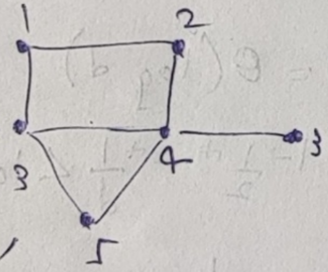
$$= H(u_0) - H(|S|)$$

Corollary:-2

Dominating Set

$V' \subseteq V$ is a Dominating set if $V = V' \cup \{$neighbou[r]

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



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

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

Roman domination:-

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