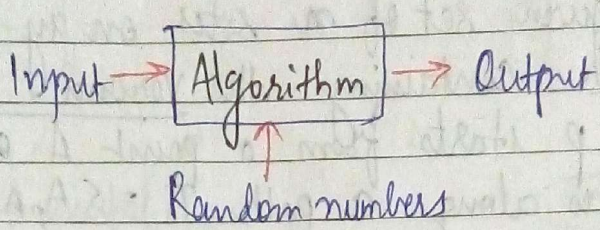
Orion

PAGE:
DATE: 11/09/17

Randomized Algorithms



1) Las Vegas algorithms
   - Exact / correct output always
   - Runtime complexity is not exact but expected.

2) Monte Carlo algorithms
   - Runtime complexity is exact
   - Output is incorrect upto a maximum probability (i.e., output is correct by a minimum probability).

Problem- Input is a 1D array $A[1 \ldots n]$ containing n distinct integers. Find $x \in A$ such that
i) At least 25% elements of A are less than x
ii) At least 25% elements of A are greater than x

1. Generate a pseudo-random number r in the interval $[1, n]$
2. $x \leftarrow A[r]$
3. if x satisfies (i) and (ii) ⟶ O(n) time
4. then report x and terminate
5. else goto step 1.

Expected number of repetitions of steps 1-5 = 2 (why?)
⇒ Expected Time Complexity = $2 \cdot O(n) = \boxed{O(n)}$

2 tosses of an unbiased coin

Sample space $\Omega = \{HH, HT, TH, TT\}$

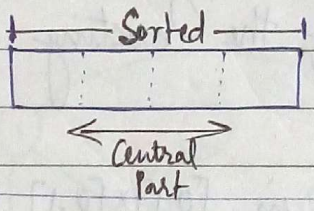
Expectation of a random variable X

$E(X) = p(X=0).0 + p(X=1).1 + p(X=2).2$ (#heads in 2 tosses) $\rightarrow \{0,1,2\}$

$= 0.\frac{1}{4} + 1.\frac{1}{2} + 2.\frac{1}{4} = \frac{1}{2} + \frac{1}{2} = 1.$

3 tosses $\rightarrow$ $E(X) = 0.\frac{1}{8} + 1.\frac{3}{8} + 2.\frac{3}{8} + 3.\frac{1}{8} = \frac{3}{2} = 1.5.$

4 tosses $\rightarrow$ $E(X) = 0 + 1.\frac{4}{16} + 2.\frac{6}{16} + 3.\frac{4}{16} + 4.\frac{1}{16} = 2$



p(a lies in Central Part) $= 1/2$

12/09/17

Closest Pair of Points – Las Vegas Algorithms (Incremental Algorithm)

Let $P = \{p_1, p_2, \ldots, p_n\}$ be the input point set

Obs 1: Define $P_i = \{p_1, p_2, \ldots, p_i\} \subseteq P$.

Define δ_i = closest-pair distance in P_i

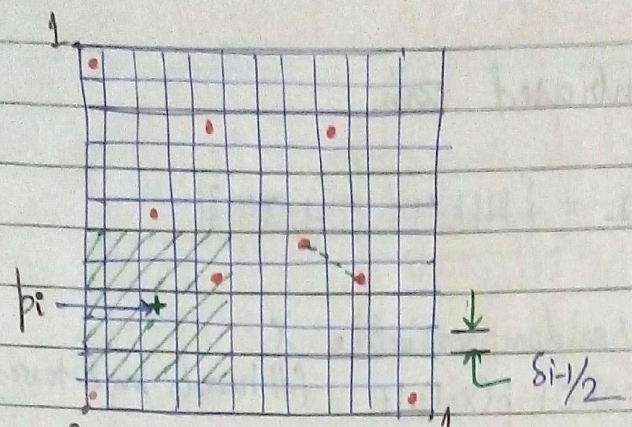
Case 1: $\delta_i = \delta_{i-1} \Rightarrow p_i$ is not a point of the closest pair in P_i

Case 2: $\delta_i < \delta_{i-1} \Rightarrow p_i$ is a point of the closest pair in P_i

Obs 2: What's the probability of Case 2?

$$\frac{i-1}{\binom{i}{2}} = \frac{2}{i}$$

$$\Rightarrow P(\delta_i < \delta_{i-1}) = \frac{2}{i}$$



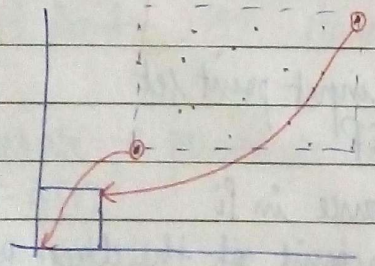
grid on a unit square containing P_{i-1}

Expected Time Complexity:

If $\delta_i < \delta_{i-1}$, then the grid is reconstructed and the points in P_i are assigned new addresses in the corresponding hash table. This takes $O(i)$ time.

Otherwise the grid remains unchanged and the point p_i is just assigned an address of the existing hash table. This takes $O(1)$ time.

Scaling the point coordinates in $[0,1] \times [0,1]$ space takes $O(n)$ time (translation & scaling down).



Divide by $\max(\Delta x, \Delta y)$.

$$O(n) + \sum_{i=3}^{n} \left(\binom{2}{i} O(i) + \left(1 - \binom{2}{i}\right) O(1) \right)$$

$$= O(n) + \sum_{i=3}^{n} O(1)$$

$$= O(n) + O(n) = O(n)$$

Orion
PAGE:
DATE: 03/10/17

MIN-CUT IN AN UNDIRECTED GRAPH

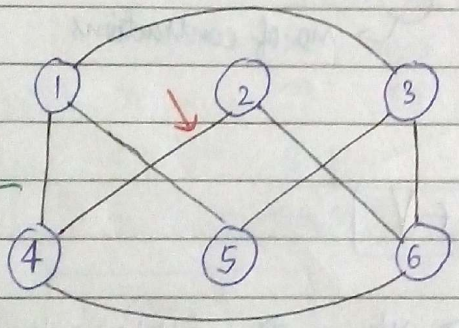
$G(V,E)$ is an undirected graph.
Let $S \subset V$ and $T \subset V$
such that $S \cup T = V$ $\quad S \cap T = \phi$.
Then (S, T) is a cut in G.
Let $E(S, T) = \{(u, v) : u \in S \text{ and } v \in T\}$
Then $|E(S, T)|$ is called the size of the cut (S, T)

Min-cut is a/the cut of smallest size.

Monte-Carlo Algorithm

It will not possibly produce the exact min-cut but some cut whose size is arbitrarily close to the size of an exact soln.

Ex.



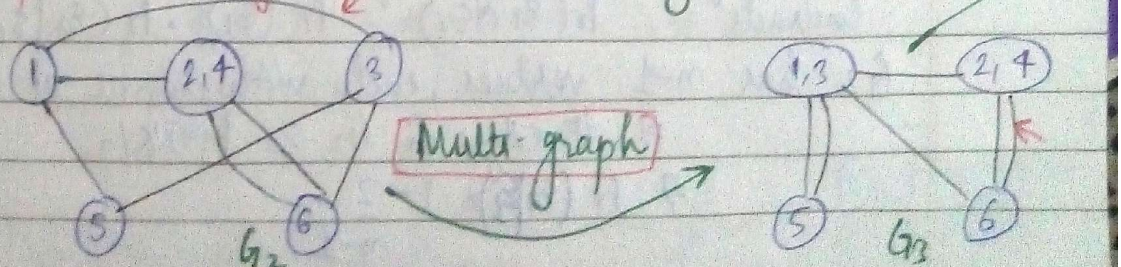
$G_1 = G$

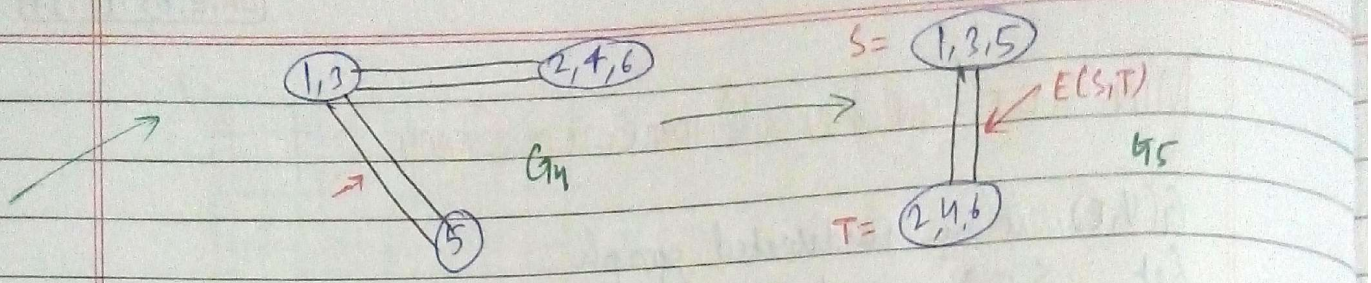
$S = \{1,2,3\}$ $\quad \Rightarrow E(S,T) = \{(1,4),(1,5),(2,4),(2,6),(3,5),(3,6)\}$
$T = \{4,5,6\}$.

$S = \{1,3,5\}$, $\quad T = \{2,4,6\} \Rightarrow E(S,T) = \{(1,4),(3,6)\}$ → Min Cut

ANUBHAV JAIN NOTES

Operation - Edge contraction (by Karger)





Observation :-

In the final graph G_{n-1}, each edge is an edge of the cut and so because it is never contracted in the process.

What is the probability that $E(S,T)$ in G_{n-1} is a min-cut?

Let C be a min-cut in $G = G_1$

We derive $Pr(C)$ in G_{n-1}

For notational simplicity, let k be the size (= # edges) of C

Let $\mathcal{E}_i$ be the event that no edge of C is contracted while obtaining G_{i+1} from G_i by the ith contract operation, where $i = 1, 2, \ldots, n-2$.

($n-2$ → no. of contractions)

$$Pr(\mathcal{E}_1) = 1 - \frac{k}{|E|}$$

* $\boxed{deg(v) \geq k \quad \forall \; v \in V}$

$$\Rightarrow \sum deg(v) = 2|E| \geq nk \Rightarrow |E| \geq nk/2$$

$$\Rightarrow \frac{k}{|E|} \leq \frac{2}{n} \Rightarrow 1 - \frac{k}{|E|} \geq 1 - \frac{2}{n}$$

$$\Rightarrow Pr(\mathcal{E}_1) \geq 1 - \frac{2}{n}$$

Probability that no edge of C is contracted in the first two contracts $= Pr(\mathcal{E}_1 \cap \mathcal{E}_2) = Pr(\mathcal{E}_1) \cdot Pr(\mathcal{E}_2 | \mathcal{E}_1)$

G_2 has $n-1$ vertices, each with degree at least k.

$\Rightarrow$ # edges in $G_2 \geq (n-1)k/2$

$\Rightarrow Pr(\mathcal{E}_2 | \mathcal{E}_1) \geq 1 - \frac{2}{n-1}$

$$\Rightarrow Pr(\mathcal{E}_1 \cap \mathcal{E}_2) \geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)$$

Prob that no edge of C is contracted in the first three contracts
$= Pr(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3) = Pr(\mathcal{E}_1 \cap \mathcal{E}_2) \cdot Pr(\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2)$

G_3 has $n-2$ vertices each of vertices of deg $\geq k$.

$$\Rightarrow Pr\left(\mathcal{E}_3 \mid \mathcal{E}_1 \cap \mathcal{E}_2\right) \geq 1 - \frac{2}{n-2}$$

$$\Rightarrow Pr\left(\bigcap_{i=1}^{3} \mathcal{E}_i\right) \geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)$$

$\Rightarrow$ Prob that C survives over all $n-2$ contracts

$$= Pr\left(\bigcap_{i=1}^{n-2} \mathcal{E}_i\right) \geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{n-(n-3)}\right)$$

$$= \frac{n-2}{n}\ \frac{n-3}{n-1}\ \frac{n-4}{n-2} \cdots \frac{2}{4}\cdot\frac{1}{3}$$

$$= \frac{2\cdot 1}{n\cdot(n-1)} = \frac{2}{n(n-1)} > \frac{2}{n^2} \quad \text{as } n \geq 2.$$

$$\Rightarrow \boxed{Pr\left(\bigcap_{i=1}^{n-2} \mathcal{E}_i\right) \geq \frac{2}{n^2}}$$

Let there be x independent runs of contracts on G (each run means $n-2$ contracts).

$P_{fail} = Pr(C \text{ does not survive in any run}) \leq \left(1-\frac{2}{n^2}\right)^x$

$0 < \frac{2}{n^2} < 1$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

$e^{-x} = 1 - x + \left(\frac{x^2}{2!} - \frac{x^3}{3!}\right) + \cdots$

$\rightarrow$ true if $x < 1$

$0 < x < 1 \Rightarrow e^{-x} > 1-x \Rightarrow (1-x)^{1/x} < 1/e$

if $\alpha = \frac{n^2}{2}$ runs, then $P_{fail} \leq \left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2}} < \frac{1}{e}$

$\Rightarrow P_{success} \geq 1 - \frac{1}{e} = 0.632$

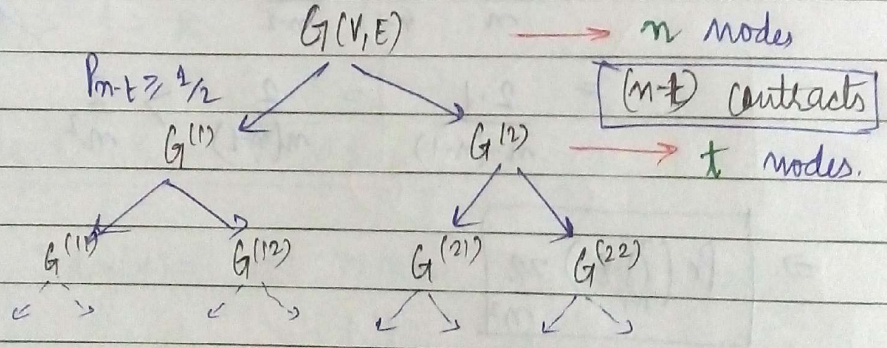
if $\alpha = n^2$ runs, $P_{fail} < \frac{1}{e^2}$ and $P_{success} \geq 1 - \frac{1}{e^2} = 0.8646$.

Tut Prob: What should be the data structure so that contraction operation can be implemented efficiently.
$\downarrow O(n)$.

05/10/17

MIN CUT - A faster Monte Carlo Algorithm

$\mathcal{E}_i$ = event that no edge of a specific min-cut C is contracted at the i^{th} step.



Let the probability that C survives after n-t contracts be P_{n-t}

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

$$= \frac{(n-2)(n-3)(n-4)\cdots(t+1)\,t\,(t-1)}{n(n-1)(n-2)\cdots(t+3)(t+2)(t+1)}$$

$$= \frac{(n-2)!/(t-2)!}{n!/t!} = \frac{t(t-1)}{n(n-1)}$$

Orion
PAGE:
DATE: / /

$$P_{n-t} \geq \frac{1}{2} \Rightarrow \frac{t(t-1)}{n(n-1)} \geq \frac{1}{2}$$

$$2t^2 - 2t \geq n^2 - n$$

$$t^2 - t - \left(\frac{n(n-1)}{2}\right) \geq 0.$$

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

$$\Rightarrow t \geq \left\lceil \frac{n}{2} \right\rceil + 1$$

$$\Rightarrow P_{n-t} \geq \frac{\left\lceil \frac{n}{\sqrt{2}} \right\rceil \left(\left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1 \right)}{n(n-1)} \geq \frac{1}{2}$$

* Probability of getting the min-cut C after the recursion.
- Let $P(n)$ denote the prob. for a graph with n nodes (or multi-graph)
- Prob that C does not survive in first $n-t$ contracts $= 1 - P_{n-t}$

$P \geq \frac{1}{2}$ Given that C survives in the first $n-t$ contracts, the prob. that it does not survive somewhere down the recursion (starting at $G^{(t)}$) $= 1 - P(t)$

$\Rightarrow$ Prob. that C does not survive in the left subtree of the recursion tree rooted at $G \leq \frac{1 - P(t)}{2}$

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

$$= P\left(\left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1\right) - \frac{1}{4}\left(P\left(\left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1\right)\right)^2$$

ANUBHAV JAIN NOTES

Orion
PAGE:
DATE: / /

By transformation of variables and using equality,

$$p(h+1) = p(h) - \frac{1}{4}\left(p(h)\right)^2 = \left(\frac{p(h)}{2}\right)^2\left(\frac{4}{p(h)} - 1\right)$$

Let $q(h) = \frac{4}{p(h)} - 1 \Rightarrow p(h) = \frac{4}{q(h)+1}$

$$\Rightarrow \frac{4}{q(h+1)+1} = \frac{1}{4} \cdot \frac{4^2}{(q(h)+1)^2} \cdot q(h) \qquad \Rightarrow q(h+1)+1 = \frac{(q(h)+1)^2}{q(h)}$$

$$\Rightarrow q(h+1) = q(h) + 1 + \frac{1}{q(h)} \qquad \text{— (1)}$$

09/10/17

Fast Min-cut Algorithm (Monte Carlo)

Last Class: $P(n)$ = prob. that a min-cut survives in a graph with n vertices.

$P(n)$ is equivalent to $p(h)$, where h is height of the recursion tree

We got: $p(h+1) = p(h) - \frac{1}{4}(p(h))^2$

Let $q(h) = \frac{4}{p(h)} - 1 \Rightarrow p(h) = \frac{4}{q(h)+1}$ — (1)

Then we got $q(h+1) = q(h) + 1 + \frac{1}{q(h)}$ — (2)

Today

$p(0) = 1$

$\Rightarrow q(0) = \frac{4}{1} - 1 = 3$

(2): $q(1) = 3 + 1 + \frac{1}{3} = 4 + \frac{1}{3} \Rightarrow 4 < q(1) < 5$

$q(2) = 4 + \frac{1}{3} + 1 + \frac{1}{4 + 1/3} \Rightarrow 5 < q(2) < 6$

Similarly, $6 < q(3) < 7$, $7 < q(4) < 8$, and so on...

Orion
PAGE:
DATE: / /

$\Rightarrow q(h) = \theta(h)$
$= \theta(\log n)$

$t \geq \left\lceil \frac{n}{2} \right\rceil + 1$

$n - t \leq ?$

① : $P(n) = p(h) = \frac{4}{\theta(\log n) + 1} = \theta\left(\frac{1}{\log n}\right)$

Time Complexity:

$$T(n) = 2T\left(\left\lceil \frac{n}{\sqrt{2}} \right\rceil + 1\right) + O(n^2)$$

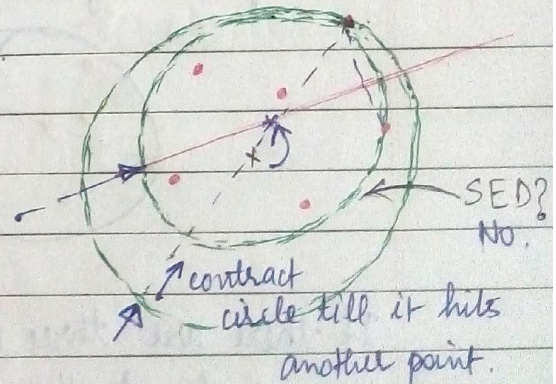
↑ for $n-t$ contracts (each contract in $O(n)$ time)

$$\boxed{T(n) = O(n^2 \log n)}$$

We can go for $O(\log n)$ independent runs to make probability constant (indep. of n).

Smallest Enclosing Disk

Brute Force:
For every 3 points, check if its circumcircle contains all other $n-3$ points in it or not.
$T(n) = {}^{n}C_3 \times (n-3) = O(n^4)$.



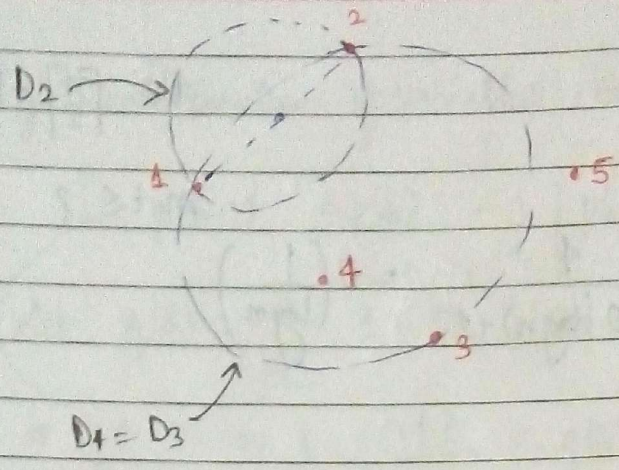
Observation 1:
At least two points lie on the boundary of SED

Two points ⇒ a diameter

ANUBHAV JAIN NOTES

Orion
PAGE:
DATE: / /

D_x: Circle/Disk containing 1st x points
C_x: Boundary of D_x.

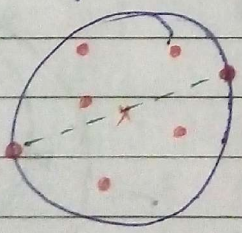


Observation2:
If $p_i \notin \mathcal{D}_{i-1}$, then $p_i \in C_i$
where $\mathcal{D}_i$ = SED of $p_1, p_2, \ldots, p_i$
and C_i = boundary of $\mathcal{D}_i$

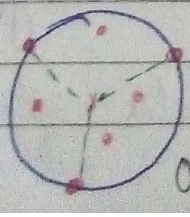
10/10/17

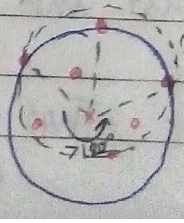
Smallest Enclosing Disk (SED)

Obs1. SED always contains atleast two points on its boundary. If there are two points on boundary, then they are the endpoints of a diameter of the SED.



If there are three (or more) points on the boundary of a SED, then each of the resultant arcs subtends atmost 180° at the centre of SED.


OK

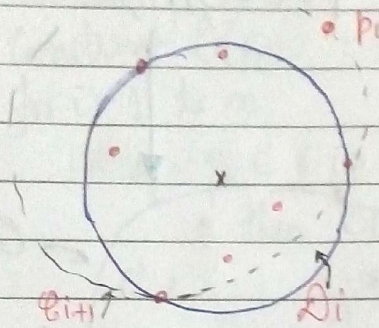


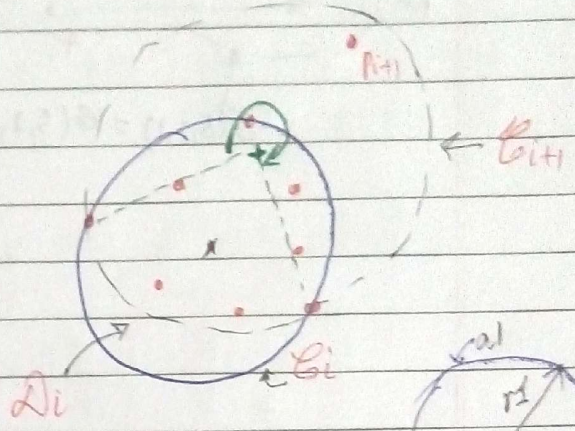
not ok

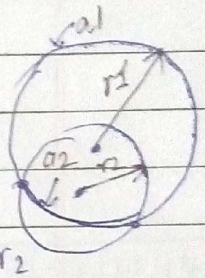
Orion
PAGE:
DATE: / /

Let D_i = SED($p_1, p_2, \ldots, p_i$) and
C_i = boundary (circle) of D_i

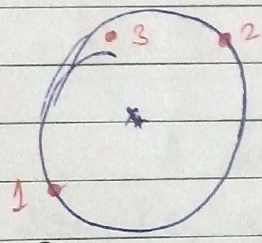
Theorem: If $p_{i+1} \in D_i$, then $D_{i+1} = D_i$
otherwise, $p_{i+1} \in C_{i+1}$





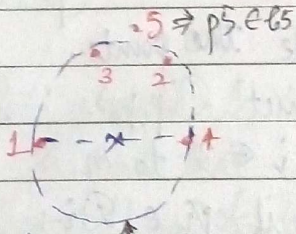


$r_1 > r_2$
$\Rightarrow a_1 > a_2$.
arcs of big circle.

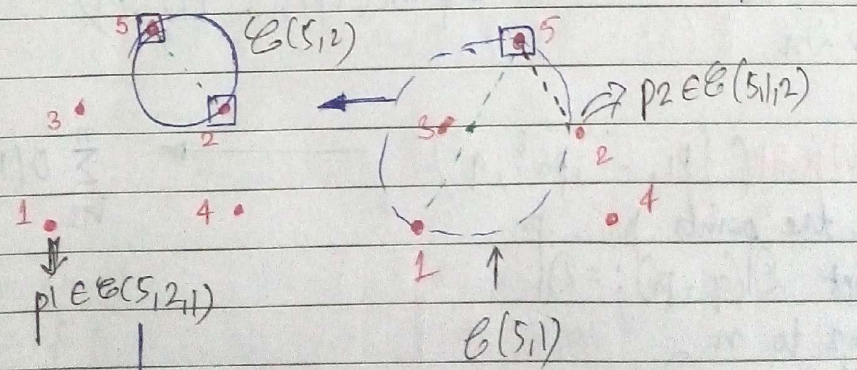


$D_3 = D_2$

4 ⇓ $p_4 \in C_4$



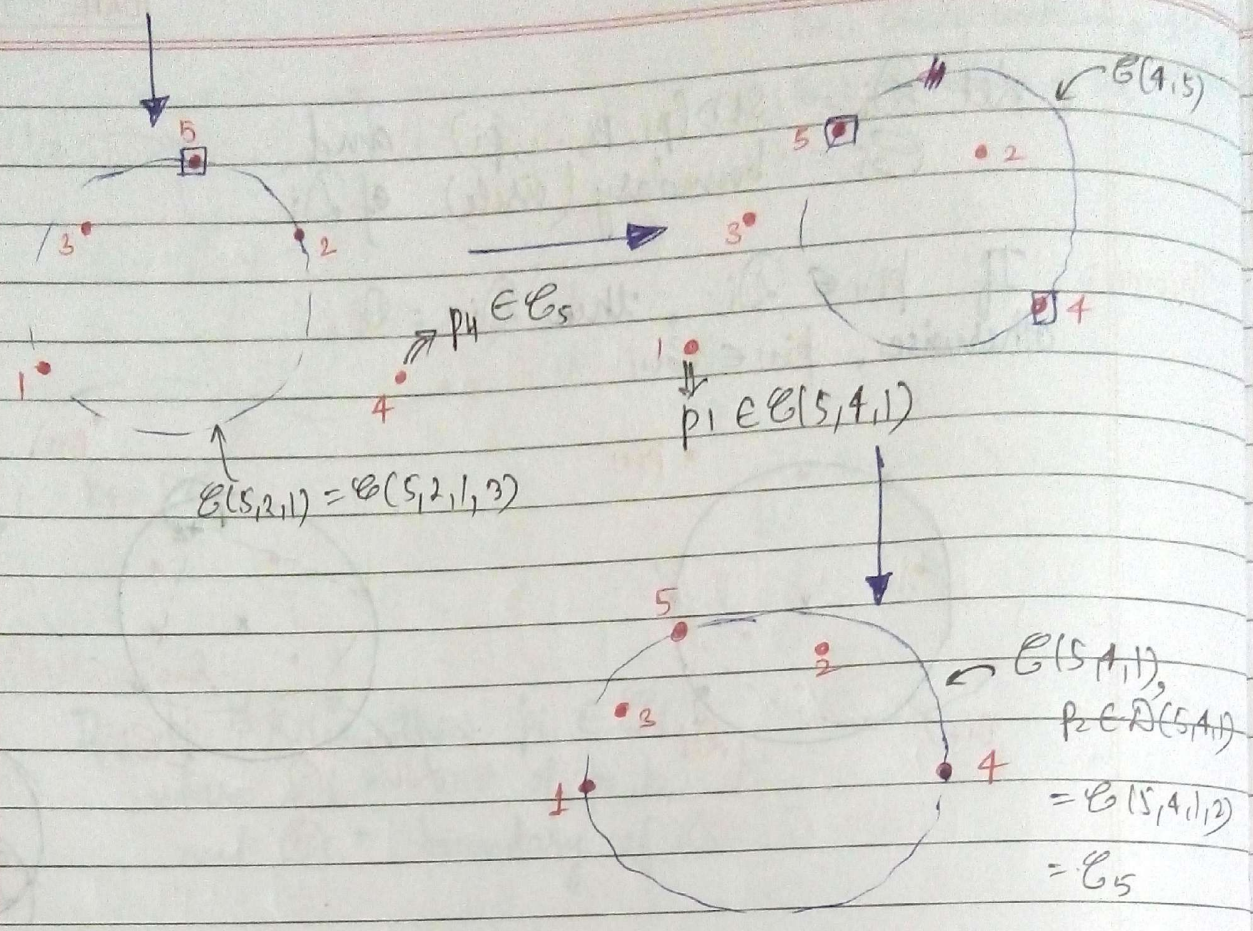
$C(1,4)$
$= C(1,4,2)$
$= C_4$



$p_1 \in C(5,2,1)$

Next page.

ANUBHAV JAIN NOTES



Algorithm SED $(\text{points } \{p_1, p_2, \ldots, p_n\})$

1. Permute the points.
2. Construct $D(p_1, p_2) := D_2$
3. for $i \leftarrow 3$ to n
4. if $p_i \in D_{i-1}$
5. then $D_i \leftarrow D_{i-1}$
6. else
7. $D_i \leftarrow$ SED with 1 point $(\{p_1, \ldots, p_i\}, p_i)$
8. return D_n

SED with 1 point $(\{p_1, \ldots, p_n\}, q)$ $\longrightarrow$ $\sum_{i=1}^{n} O(1/i) O(i) \Rightarrow O(n)$

1. Permute the points $p_1 \ldots p_n$
2. Construct $D(q, p_1) := D_1$
3. for $i \leftarrow 2$ to n
4. if $p_i \in D_{i-1}$
5. then $D_i \leftarrow D_{i-1}$

Orion
PAGE:
DATE: / /

6. else
7. $D_i \leftarrow$ SED with 2 points $(\{p_1, p_2, \ldots, p_i\}, p_i, q)$
8. Return D_n

Pr(called if $p_i \in B_i$) $= O(1/i)$

SED with 2 points $(\{p_1, \ldots, p_n\}, p, q)$ $\longrightarrow O(n)$

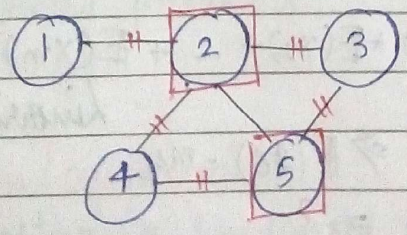
1. Permute $p_1 \ldots p_n$
2. Construct $D(p, q) := D_0$
3. for $i \leftarrow 1$ to n
4. if $p_i \in D_{i-1}$
5. then $D_i \leftarrow D_{i-1}$
6. else
7. $D_i \leftarrow D(p, q, p_i)$
8. return D_n

16/10/17.

Max-Cut

= A cut in an undirected graph $G(V, E)$ of maximum size, i.e, a partition (X, Y) of V such that the number of edges in $C = \{(x, y) : x \in X, y \in Y\}$ is of maximum cardinality.

Max-cut finding is an NP-complete problem.



$X = \{2, 5\}$ $Y = \{1, 3, 4\}$ if $X = \{1, 5\}$, $\#C = 4$.

$C = \{(1,2), (2,3), (2,4), (3,5), (4,5)\}$.

$\#C = 5$

Orion
PAGE:
DATE: / /

Max-cut – Monte Carlo Algorithm

1. for each $v \in V$
2. $\quad r \leftarrow$ random $\{0,1\}$
3. $\quad$ if $r=1$ then put v in X
4. $\quad$ else put v in Y
5. return (X, Y)

Expected size of the cut C returned by the Monte Carlo algorithm:

Each edge of C has one vertex in X and the other vertex in Y

Let $(u,v) \in E$

$Pr((u,v) \in C) = 1/2$

Let there be $|E| = m$ edges in G

Define X_i as a random variable for the i^{th} edge in E, $i = 1, 2, \ldots, m$

where $X_i = 1$ if the i^{th} edge belongs to C

$= 0$ otherwise

$$E(X_i) = 1 \cdot Pr(X_i = 1) + 0 \cdot (Pr(X_i = 0))$$
$$= 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}.$$

Define $X = X_1 + X_2 + \ldots + X_m$ as another random variable

$$\Rightarrow E(X) = E(X_1) + E(X_2) + \ldots + E(X_m) \text{ by Linearity of Expectation}$$
$$= \frac{m}{2} \Rightarrow E(\#C) = m/2$$

$Pr(\#C \geq m/2) = ?$

Let $Pr(X \geq \lceil m/2 \rceil) = p$

Then $Pr(X < \lceil m/2 \rceil) = 1 - p$

Orion
PAGE:
DATE: / /

Now, $X = \{0, 1, 2, \ldots, m\}$

$$\Rightarrow \mathbb{E}(X) = \sum_{i=0}^{m} i \cdot Pr(X=i)$$

$$= \sum_{i=1}^{\lceil m/2 \rceil - 1} i \cdot Pr(X=i) + \sum_{i=\lceil m/2 \rceil}^{m} i \cdot Pr(X=i)$$

$$\mathbb{E}(X) \leq \left(\left\lceil \frac{m}{2} \right\rceil - 1\right) \underbrace{\sum_{i=1}^{\lceil m/2 \rceil - 1} Pr(X=i)}_{Pr(X < \lceil m/2 \rceil) = 1-p} + (m) \underbrace{\sum_{i=\lceil m/2 \rceil}^{m} Pr(X=i)}_{p}$$

$$= \left(\left\lceil \frac{m}{2} \right\rceil - 1\right)(1-p) + mp$$

$$= \left\lceil \frac{m}{2} \right\rceil - p\left\lceil \frac{m}{2} \right\rceil - 1 + p + mp \geq \mathbb{E}(X) = m/2.$$

$$p + \left\lfloor \frac{m}{2} \right\rfloor p \geq 1 - \left\lceil \frac{m}{2} \right\rceil + \frac{m}{2} \geq \frac{1}{2}$$

$$p + \frac{m}{2} p \geq \frac{1}{2}$$

$$p \geq \frac{1}{2(1+m/2)} = \frac{1}{2+m}$$

⇒ Iterate the algorithm $(2+m)$ times to get a constant expected probability.

17/10/17

Derandomization

Converting a randomized algorithm to an ordinary deterministic algorithm (without compromising with the quality/efficiency).

Orion
PAGE:
DATE: / /

Max-Cut

Based on conditional expectance
$G = (V, E)$ is the input graph.

$V = \{v_1, v_2, \ldots, v_n\}$
Let $V_i = \{v_1, v_2, \ldots, v_i\}$, where $i < n$

Let (X_i, Y_i) be the cut obtained for V_i.

Let $\mathbb{E}(K_{(X,Y)})$ denote the expected size of the final cut.
Let $\mathbb{E}(K_{(X_i,Y_i)})$ " " " " " the cut (X_i, Y_i)

$$\mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i)) = \frac{1}{2}\,\mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i) \wedge v_{i+1} \in X) + \frac{1}{2}\,\mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i) \wedge v_{i+1} \in Y)$$

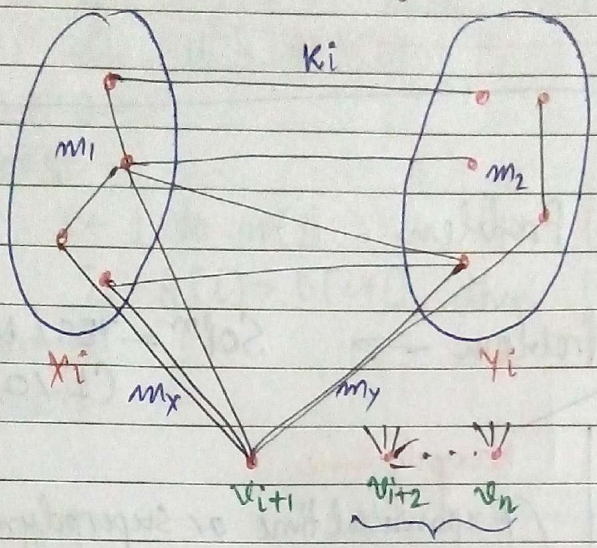
$$\Rightarrow \max\{\mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i) \wedge v_{i+1} \in X), \mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i) \wedge v_{i+1} \in Y)\} \geq \mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i)) \quad -(1)$$

Hence, as per Eq (1), we put v_{i+1} in X or in Y depending on whether $\mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i) \wedge v_{i+1} \in X)$ is larger than $\mathbb{E}(K_{(X,Y)} \mid (X_i, Y_i) \wedge v_{i+1} \in Y)$ or not.

This is a deterministic step.
So if (X_i, Y_i) is obtained deterministically, then the subsequent cuts (X_{i+1}, Y_{i+1}), (X_{i+2}, Y_{i+2}), ..., $(X, Y) = (X_n, Y_n)$ are also obtained deterministically.

Orion
PAGE:
DATE: / /

How to determine the larger cut between $(x_i \cup \{v_{i+1}\}, y_i)$ and $(x_i, y_i \cup \{v_{i+1}\})$



Let
- k_i = # crossing edges
- m_x = #edges with one vertex in x_i & other v_{i+1}
- m_y = # edges with one vertex in y_i & other v_{i+1}
- m_1 = # edges fully lying in x_i
- m_2 = # edges fully lying in y_i

Let $m = |E|$

Let m' = # edges whose atleast one vertex is in $\{v_{i+2}, \ldots, v_n\}$

$$\Rightarrow m' = m - (k_i + m_x + m_y + m_1 + m_2)$$

$$v_{i+1} \in X \Rightarrow \mathbb{E}(K_{(X,Y)} \mid (x_i, y_i)) = k_i + m_y + m'/2$$

$$v_{i+1} \in Y \Rightarrow \mathbb{E}(K_{(X,Y)} \mid (x_i, y_i)) = k_i + m_x + m'/2$$

$\Rightarrow$ if $m_y > m_x$, then $v_{i+1} \in X$
else $v_{i+1} \in Y$.

$$\mathbb{E}(K_{(X,Y)} \mid (x_1, y_1)) \leq \mathbb{E}(K_{(X,Y)} \mid (x_2, y_2)) \leq \ldots \leq \mathbb{E}(X, Y)$$

Derandomized algorithm for max-cut

```
1. X ← {v1}, Y ← {}
2. for i ← 2 to n
3.     mx ← |{(u, vi) : u ∈ X}|   } → O(|Adj[vi]|)
4.     my ← |{(u, vi) : u ∈ Y}|   }
5.     if (mx ≤ my) then X ← X ∪ {vi}
6.     else Y ← Y ∪ {vi}
7. return (X, Y)
```

ANUBHA JAIN NOTES

Orion
PAGE:
DATE: / /

$$\Rightarrow T = \sum_{i=1}^{n} O(|Adj[v_i]|) = O(E)$$

Optimization Problem

↓

Decision Problem → Sol^n = YES / NO (1 / 0)

Poly time	Non poly time
$T(n) \leq K n^p$ where K, p are consts.	(Exponential time or superpolynomial time) e.g. $2^{f(n)}$

Non deterministic Algorithms

Based on three procedures -

- SUCCESS = successful termination of the algorithm
- FAILURE = unsuccessful termination
- SELECT = arbitrary selecting one of the possible options in favour of the solution

---

NON-DET-SEARCH (Key x, list A[1..n])

1. i ← SELECT ({1, 2, ..., n})
2. if x = A[i] then SUCCESS
3. else FAILURE

T(n) = O(1)

NON-DET-SORT (list A[1..n]) // all elements are positive

1. for i ← 1 to n
2. B[i] ← 0
3. for i ← 1 to n

Orion

PAGE:
DATE: / /

```
4.      j ← SELECT({1,2,...,n})
5.      if B[j] ≠ 0 then FAILURE
6.      else B[j] ← A[i]
```

produces a permutation of A[1..n]

~~7. report B~~
~~8. SUCCESS~~

```
7.  for i ← 1 to n-1
8.      if B[i] > B[i+1] then FAILURE
9.  report B
10. SUCCESS
```

verifies whether B[1..n] is sorted in inc. order.

$\mathcal{P}$ = class of problems that can be solved by deterministic algorithms in polynomial time.

$\mathcal{NP}$ = class of problems that can be solved by non-deterministic algorithms in polynomial time.

$\mathcal{P} \subseteq \mathcal{NP}$

? (Billion $ ques)

23/10/17

NP-hard class of problems (NPH)
NP-complete class of problems (NPC)

Let X be a problem.

1) $X \in NPC \Rightarrow X \in NP$
$\not\Leftarrow$

Satisfiability Problem (SAT) $\in$ NPC

2) $X \in NPC \Rightarrow SAT \leq_p X$

(SAT can be reduced to X in polynomial time)

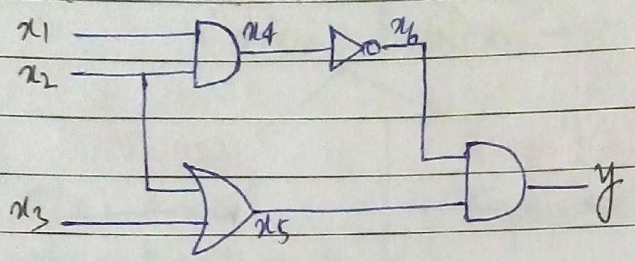
These are two necessary & sufficient conditions to prove that a problem X is NP-Complete.

ANUBHAV JAIN NOTES

Orion
PAGE:
DATE: / /

Circuit Satisfiability Problem (SAT)

Given a boolean circuit made of AND, OR, NOT gates and n input wires, we need to find whether there exists an input certificate $\langle x_1, x_2, \ldots, x_n \rangle$ for which the circuit has output = 1 (TRUE)



Cook's Theorem

SAT $\in$ P iff P = NP

Formula Satisfiability Problem (ϕ-SAT)

$$\phi = \big((x_1 \vee \neg x_2) \wedge x_3\big) \vee \big((\neg x_1 \wedge x_2) \wedge \neg x_3\big)$$

To prove: that SAT is NP Complete, we need to prove:

1) An input certificate can be verified in poly time.
2) Any instance of SAT can be reduced to some instance of ϕ-SAT s.t. a solution to that instance of ϕ-SAT corresponds to a solution of the (any) given instance of SAT.

$$\phi = (x_1 \wedge x_2 \leftrightarrow x_4) \wedge (x_2 \vee x_3 \leftrightarrow x_5) \wedge (\neg x_4 \leftrightarrow x_6) \wedge (x_5 \wedge x_6 \leftrightarrow y) \wedge y$$

Observe that a YES-instance of ϕ is also a YES instance of the given circuit and vice-versa. Hence the correspondance

Orion
PAGE:
DATE: 24/10/17

NP - Complete Problems

$SAT \leq_p \phi\text{-SAT} \leq_p 3\text{-SAT}$

3-SAT: A boolean formula is given in the form $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$ where each C_i $(1 \leq i \leq k)$ is of the form $C_i = y_{i_1} \vee y_{i_2} \vee y_{i_3}$, where the literals y_{i_p} $(p = 1,2,3)$ are either the input binary variables $(x_1, x_2, \ldots x_n)$ or their negations $(\neg x_1, \neg x_2, \ldots, \neg x_n)$
We have to decide whether there exists an input certificate for which ϕ is TRUE.

Ex: $\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_3)$

Here, $n = 3$, $k = 3$

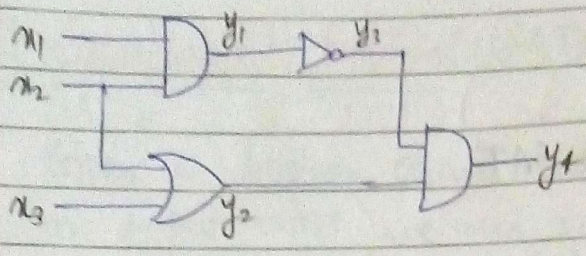
ϕ is in 3-CNF (Conjunctive Normal Form)

$k = \text{poly}(n)$

3-SAT $\in$ NP, as any input certificate can be verified in $O(k)$ time, which is a polynomial of n.

x	y	$x \leftrightarrow y$
T	T	T
T	F	F
F	T	F
F	F	T

Now, we prove that $SAT \leq_p 3\text{-SAT}$
For this, we take an instance of circuit and construct its equivalent 3-CNF formula in poly time.



1) Label all wires.
2) Construct the equivalent formala. This is a CNF but not a 3-CNF.
(conjuctive form)

$$\phi = (x_1 \wedge x_2 \leftrightarrow y_1) \wedge (x_2 \vee x_3 \leftrightarrow y_2) \wedge (\neg y_1 \leftrightarrow y_3) \wedge (y_2 \wedge y_3 \leftrightarrow y_4) \wedge y_4$$

3) Consider each clause one by one and convert it into 3-CNF.
We use truth table for this.
For every clause, we need $O(1)$ time.

$C_1 = x_1 \wedge x_2 \leftrightarrow y_1$

x_1	x_2	y	C_1	
0	0	0	1	
0	0	1	0	←
0	1	0	1	
0	1	1	0	←
1	0	0	1	
1	0	1	0	←
1	1	0	0	←
1	1	1	1	

$\neg C_1 = (\neg x_1 \wedge \neg x_2 \wedge y_1) \vee$
$(\neg x_1 \wedge x_2 \wedge y_1) \vee$
$(x_1 \wedge \neg x_2 \wedge y_1) \vee$
$(x_1 \wedge x_2 \wedge \neg y_1)$

$\Rightarrow C = \neg(\neg C)$
$= (x_1 \vee x_2 \vee \neg y_1) \wedge (x_1 \vee \neg x_2 \vee \neg y_1) \wedge$
$(x_1 \vee x_2 \vee \neg y_1) \wedge (\neg x_1 \vee \neg x_2 \vee y_1)$
– by DeMorgan's Thm.

$C_3 = \neg y_1 \leftrightarrow y_3$

y_1	y_3	C_3	
0	0	0	←
0	1	1	
1	0	1	
1	1	0	←

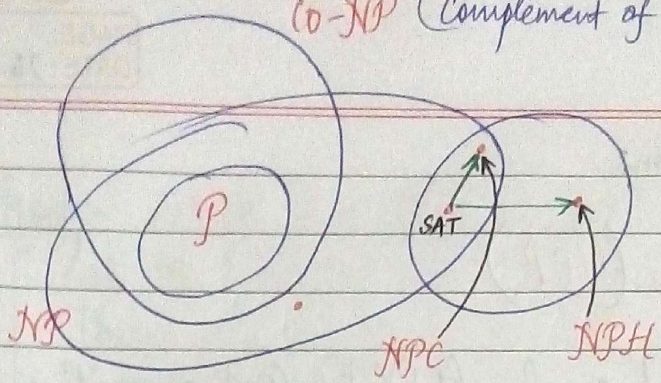
$\neg C_3 = (\neg y_1 \wedge \neg y_3) \vee (y_1 \wedge y_3)$

$\Rightarrow C_3 = (y_1 \vee y_3) \wedge (\neg y_1 \vee \neg y_3)$

Use a dummy variable z.
Then we can write

$C_3 = (y_1 \vee y_3 \vee z) \wedge (y_1 \vee y_3 \vee \neg z) \wedge$
$(\neg y_1 \vee \neg y_3 \vee z) \wedge (\neg y_1 \vee \neg y_3 \vee \neg z)$

$(y_1 \vee y_3) = (y_1 \vee y_3) \vee 0$
$= (y_1 \vee y_3) \vee (z \wedge \neg z)$
$= (y_1 \vee y_3 \vee z) \wedge (y_1 \vee y_3 \vee \neg z)$



Clique Decision Problem (CDP)

A subgraph $G'(V', E')$ of an undirected graph $G(V, E)$ in which every pair of vertices has an edge is called a clique.

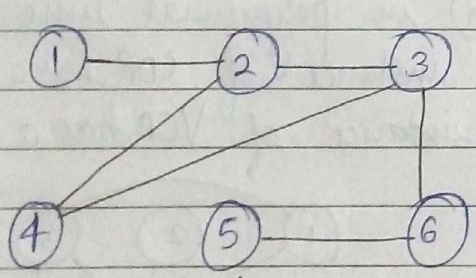
$V' \subseteq V$, $E' \subseteq E$, $(a,y) \in E'$ $\forall (a,y) \in V' \times V'$

- Finding the largest clique is an optimization problem.
- Determining whether there exists a clique of a given size k is a decision problem (CDP).

To prove that CDP $\in$ NPC, we show:

1) CDP $\in$ NP
2) 3-SAT $\leq_p$ CDP

Ex.



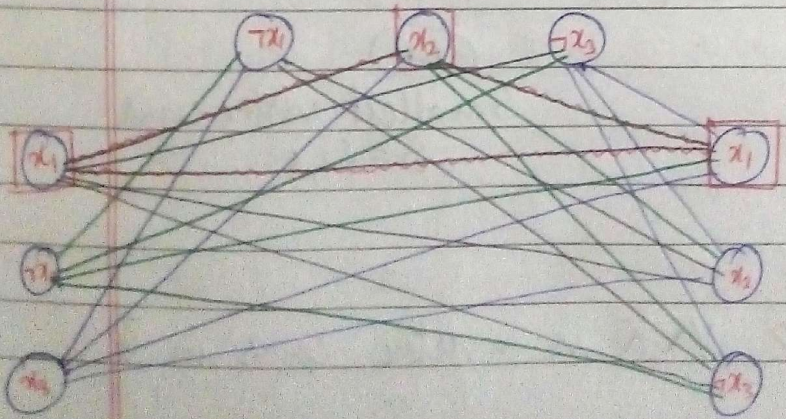
Here $k \leq 3$.

$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$

$\phi(1,1,1) =$ TRUE.

$\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$.

In $G(V, E)$, if there is a clique of size k, then ϕ is satisfiable



V = all literals in ϕ

$E = \{(x,y) : (x \neq \neg y) \wedge (\text{clause}(x) \neq \text{clause}(y))\}$

Orion
PAGE:
DATE: 26 / 10 / 17

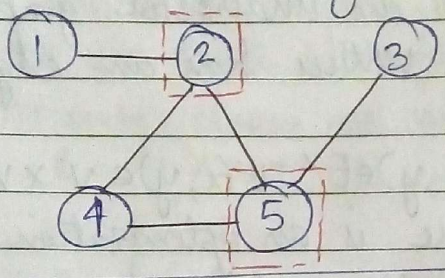
NP Complete Problems

Vertex Cover Problem (VCP)

Decision version -

Given an undirected graph $G(V, E)$ and a positive integer s, determine whether there exists $S \subseteq V$ of size s such that for every edge (u, v) in E, at least one between u and v belongs to S.

Ex.



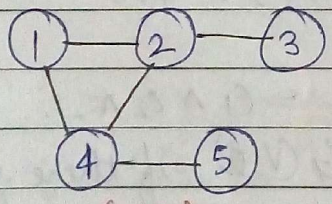
$S = \{2, 5\}$ is the vertex cover of smallest size.

→ 1) $VCP \in NP$

2) $CDP \leq_p VCP$

To show that $CDP \leq_p VCP$, we consider an instance of VCP (by a deterministic algorithm) in polynomial time, and their claim that the given instance of CDP has a solution if and only if the reduced instance of VCP has a solution.

CDP, reduce it to some instance of VCP



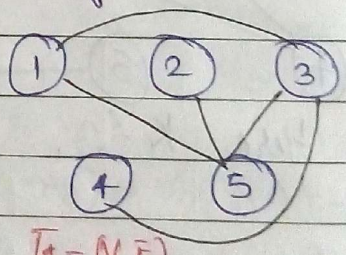
$G(V, E)$

CDP

Largest clique

$K = \{1, 2, 4\}$

$\bar{E} = V^2 \setminus E$



$\bar{G} = (V, \bar{E})$

VCP

Smallest vertex cover

$S = \{3, 5\}$

$\bar{G}$ = complement of G

K is a clique in $G \Leftrightarrow V \setminus K$ is a vertex cover in $\bar{G}$

Orion
PAGE:
DATE: / /

Proof:- Let K is a clique in G

Let $u \in K$ and $v \in V \setminus K$ such that $(u,v) \notin E$.
Then $(u,v) \in \bar{E}$
So every vertex u in K is covered by some vertex v in $V \setminus K$ (in $\bar{G}$). This means $V \setminus K$ is a vertex cover in $\bar{G}$.

Let $(u,v) \in K^2$
Then $(u,v) \notin \bar{E}$
$\Rightarrow$ (u,v) need not be covered in $\bar{G}$.
$\Rightarrow$ $V \setminus K$ suffices as a vertex cover in $\bar{G}$.

30/10/17.

$CDP \leq_p VCP$

$G(V,E)$ $\quad$ $\bar{G}(V,\bar{E})$
is an input instance

Clique K in G $\Leftrightarrow$ Vertex cover $S = V \setminus K$ in $\bar{G}$

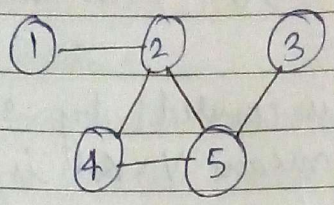
Proof:-
$V \setminus K$ is a vertex cover in $\bar{G} \Rightarrow$ K is a clique in G

Pf:-
Let u and v be (any) two vertices such that $u \notin V \setminus K$ and $v \notin V \setminus K$. Then $(u,v) \notin \bar{E}$
$\Rightarrow (u,v) \in E$

All such vertex pairs are in E
$\Rightarrow$ They form a clique (K) in G.

SAT $\rightarrow$ ϕ-SAT $\rightarrow$ 3SAT $\rightarrow$ CDP $\rightarrow$ VCP $\rightarrow$ ISP

Independent Set Problem (ISP)



$I = \{1, 3, 4\}$.

An independent set in an undirected graph $G(V, E)$ is a subset I of V in which there is no edge.

Given a size m and an undirected graph G, determine whether G has an independent set of size m.

$G \rightarrow \bar{G}$

(V, E) (V, E)

K is clique in $G \Leftrightarrow$ K is I.S in $\bar{G}$

clique K $\Leftrightarrow$ $V \setminus K$ is vertex cover in $\bar{G}$

V.C and I.S are complement to each other in the same graph.

Set Cover Problem (SCP)

Given a universal set $\mathcal{U} = \{x_1, x_2, \ldots x_n\}$ and a family of subsets of $\mathcal{U}$, $\mathcal{F} = \{S_1, S_2, \ldots S_k\}$, and a positive integer m, determine whether there exist m members of $\mathcal{F}$ (each member here is some S_i in $\mathcal{F}$) such that their union produces $\mathcal{U}$.

Ex. $\mathcal{U} = \{1, 2, \ldots 7\}$ $\mathcal{F} = \{S_1 = \{2,5,6\}, S_2 = \{1,2,3\}, S_3 = \{3,4,7\}, S_4 = \{1,2,5,7\}, S_5 = \{2,4,7\}\}$,

$m = 3$.

YES $= \{S_1, S_2, S_3\}$.

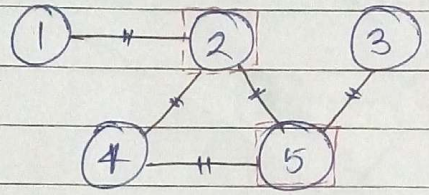
Orion

PAGE:
DATE: 31 / 10 / 17

Set Cover Problem (SCP)

Statement - Last Class

$VCP \leq_p SCP$



$V' = \{2,5\}$ is a vertex cover of size 2.
$\{1,4,3\}$ is not a vertex cover.

Reduction from VCP to SCP

Let $G(V,E)$ be an input instance of VCP

Define $\mathcal{U} = \{(u,v) : (u,v) \in E\}$

$\Rightarrow |\mathcal{U}| = |E| = n$ (say)

Define $\mathcal{F} \{\{u_i, v_i\} : (u,v) \in E \wedge 1 \leq i \leq |V| = k \text{ (say)}\}$

$\Rightarrow |\mathcal{F}| = |V| = k$

$\mathcal{U} = \{(1,2), (2,4), (2,5), (3,5), (4,5)\}$

$\mathcal{F} = \{\{(1,2)\}, \{(1,2),(2,4),(2,5)\}, \{(3,5)\}, \{(2,4),(4,5)\}, \{(2,5),(3,5),(4,5)\}\}$

$\mathcal{F}$ =

$\mathcal{U}$ = E	(1,2)	(2,4)	(2,5)	(3,5)	(4,5)
v = 1	1	0	0	0	0
2	1	1	1	0	0
3	0	0	0	1	0
4	0	1	0	0	1
5	0	0	1	1	1

Incidence Matrix

Orion
PAGE:
DATE: / /

VCP $\leq_p$ SCP: Every instance of VCP can be reduced to some instance of SCP but every instance of SCP cannot necessarily be reduce to some instance of VCP.

Hitting Set Problem (HSP)

$\mathcal{U} = \{x_1, x_2, \ldots, x_n\}$

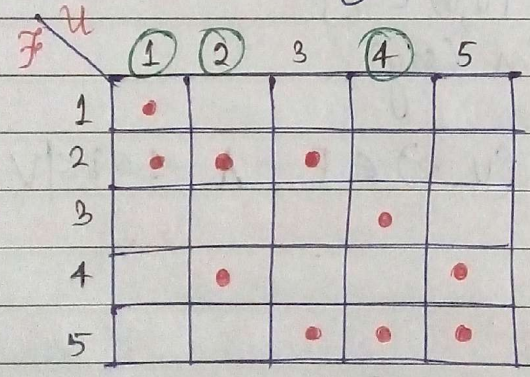
$\mathcal{F} = \{S_1, S_2, \ldots, S_k\}$

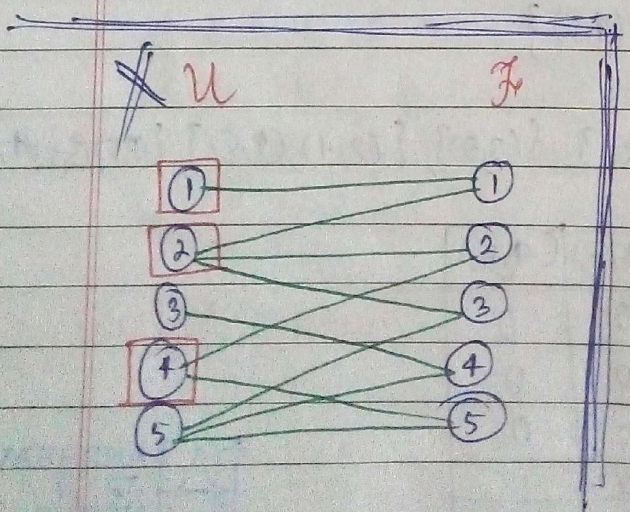
$m \leq n$

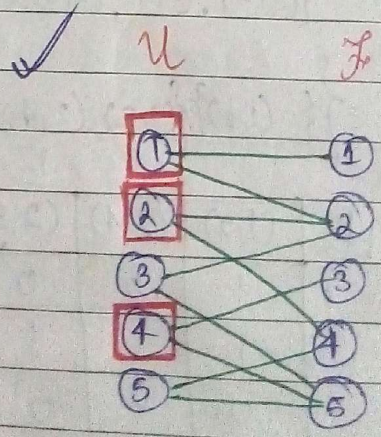
Decide whether there exists a subset S of $\mathcal{U}$ containing m elements such that

$S \cap S_i \neq \phi$ for $i = 1, 2, \ldots, k$

(S is called hitting set)

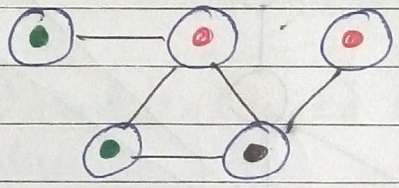






Graph Coloring Problem

Given an undirected graph $G(V, E)$ and a natural number k, decide whether G can be colored by k different colors such that no two adjacent vertices get the same color.

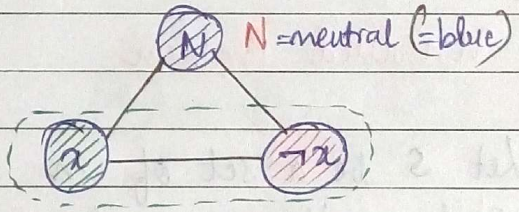


widgets

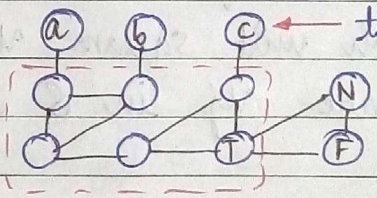
$3\text{-SAT} \leq_p 3\text{-COLORING}$

instance is ϕ in 3-CNF

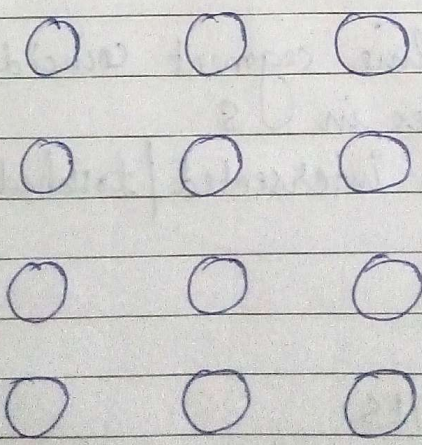
variable gadget:



clause gadget:



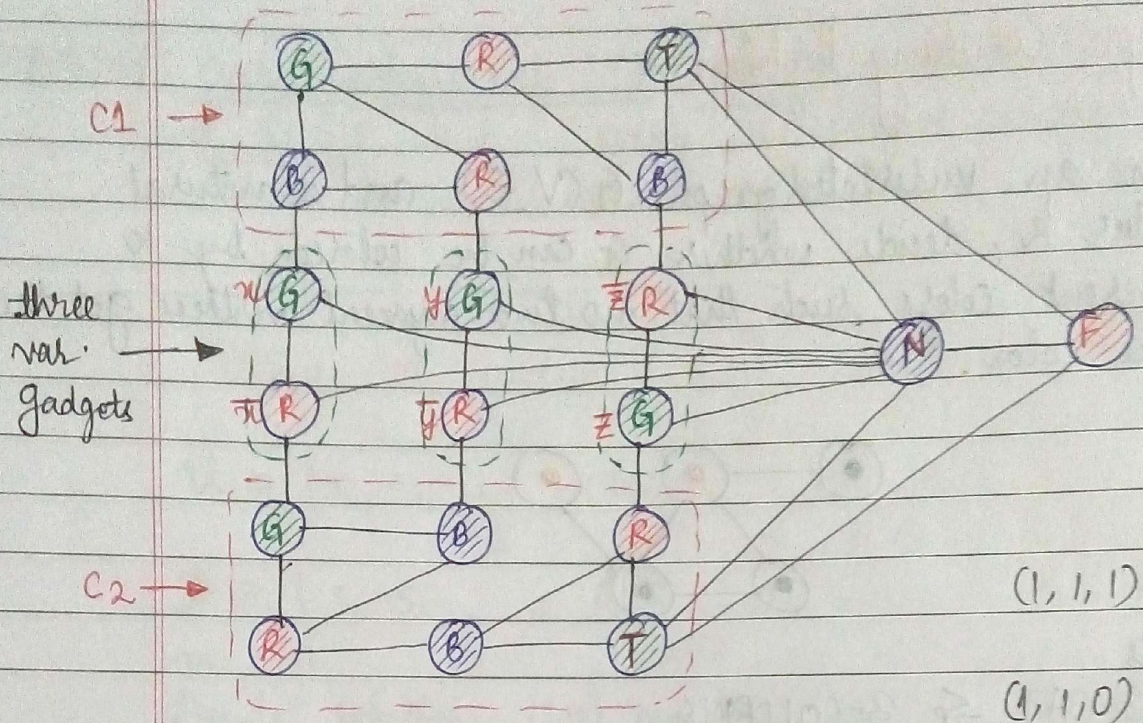
Ex. $\phi = (x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z)$



(Next Page)

Orion
PAGE:
DATE: / /

N ⇒ B
T ⇒ G
F ⇒ R



(1, 1, 1) → TRUE

(1, 1, 0)
↳ conflict

Unsolved Problem

02/11/17

Let S be a set of n unit squares such that:

(i) Each unit square has centre with integer coordinates.

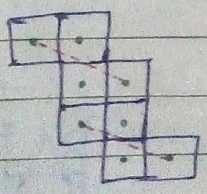
(ii) For each unit square u_i, there exists at least another unit square u_j in S such that u_i & u_j have a common side.

We have to decide whether there exists a set L of k line segments such that:

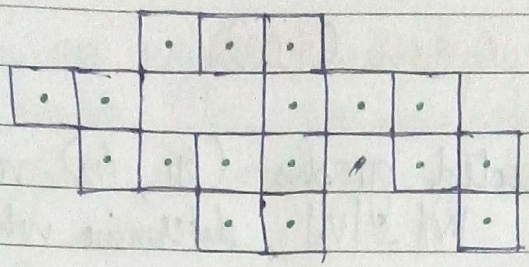
(i) → Both the endpoints of each line segment coincide with the

(ii) → centers of two unit squares in S

↳ Each unit square in S is intersected / touched by at least one line segment of L



n = 8 → k = 2 → YES

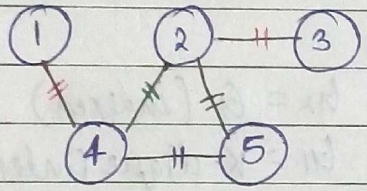


Tutorial

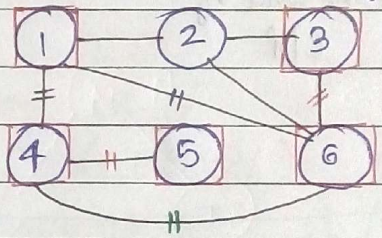
Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two undirected graphs with labeled vertices.

Let $|V_1| \leq |V_2|$.

Decide whether G_1 is isomorphic to some subgraph of G_2.



G_1



G_2

f: $V_1 \rightarrow V_2$ such that

$$(x, y) \in E_1 \Leftrightarrow (f(x), f(y)) \in E_2$$

Prove that this is an NP-Complete Problem.

Orion
PAGE:
DATE: 06/11/17

Subgraph Isomorphism (SIP)

Given two undirected graphs (G_1, G_2) where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $|V_1| \leq |V_2|$, determine whether G_1 is a subgraph induced by G_2

It is an NP-Complete problem.

Proof:- 1) It is NP (why?)

2) Clique Decision Problem (CDP) $\leq_p$ Subgraph Isomorphism Prob. (SIP)

G, K — instance of CDP ⟶ G_1, G_2 — instance of SIP

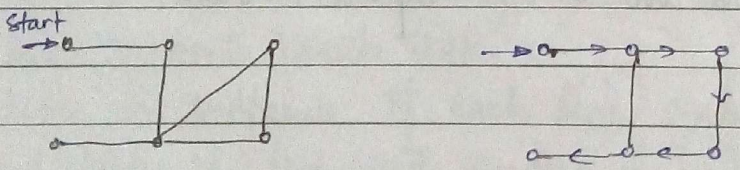
$G_2 = G$ (indexed)

$G_1 =$ K-clique (indexed)

Common

Hamiltonian Path and Cycle

Given an undirected graph $G(V, E)$, determine whether there exists a path (HAM Path) in which every vertex is visited exactly once.



NO HAM Path. YES

HAM cycle = HAM Path + an edge from the last to the start vertex in the path.

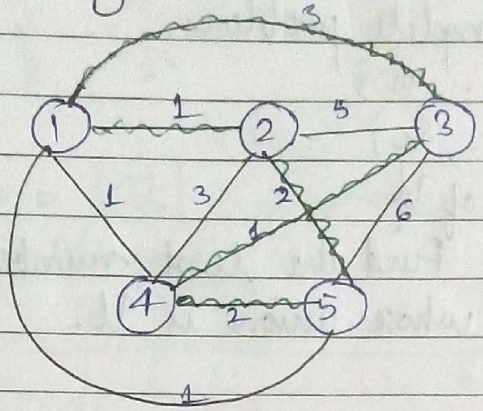
HAM Cycle Problem is NP Complete

⇒ Travelling SalesMan Problem (TSP) is NP Complete

Orion

PAGE:
DATE: / /

TSP: Given a weighted, undirected, complete graph $G(V, E)$ and given a number c, determine whether G contains a cycle of total weight c in which every vertex is visited exactly once.



$1 \to 2 \to 5 \to 4 \to 3 \to 1$

Total weight $= 1 + 2 + 2 + 1 + 3$
$= 9.$

$HCP \leq_p TSP$

HCP		TSP
G	$\longrightarrow$	wt. G'
(V, E)		$(V, E = V \times V)$

$w(u, v) = 0$ if $(u, v) \in E$
$= 1$ otherwise

SAT $\rightleftarrows$ ϕ-SAT $\rightleftarrows$ 3-SAT $\longrightarrow$ 3-COLOR

3-SAT $\downarrow$ CDP

CDP $\downarrow$ VCP $\nearrow$ Subset Sum

VCP $\longleftarrow$ HCP, VCP $\longrightarrow$ SCP (Set Cover)

HCP $\downarrow$ TSP

VCP $\updownarrow$ ISP

SCP $\downarrow\uparrow$ Hitting Set

Orion
PAGE:
DATE: 7/11/17

Approximation Algorithms

Set Cover Problem

This is an NP-Complete problem.

$\mathcal{U} = \{x_1, x_2, \ldots, x_n\}$

$\mathcal{F} = \{S_1, S_2, \ldots, S_k\}$

each S_i is a subset of $\mathcal{U}$

Optimization problem: Find the least number of subsets (i.e, members of $\mathcal{U}$) whose union is $\mathcal{U}$.

Ex. $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7\}$.

$S_1 = \{3, 5, 7\}$

→ $S_2 = \{1, 2, 4, 6, 7\}$ Choose set with max. elements.

$S_3 = \{1, 2, 4, 5\}$

$S_4 = \{3, 5\}$

$S_5 = \{2, 5, 7\}$.

$S_1 \cup S_2 = \mathcal{U}$

⇒ Set Cover = $\{S_1, S_2\}$

Minimization Problem

x_{opt} = optimal soln.

x_{app} = approx soln.

⇒ $x_{app} \geq x_{opt}$.

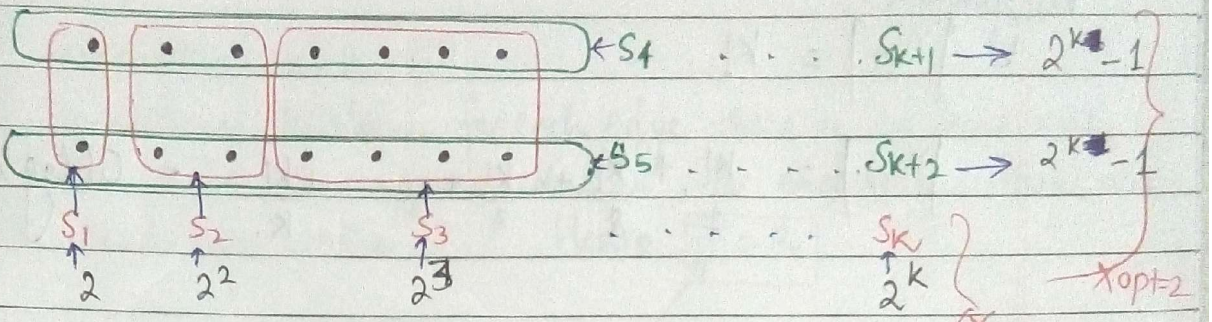
Let $x_{app} = (1 + \varepsilon)\, x_{opt}$,

where $\varepsilon \geq 0$.

⇒ $\frac{x_{app}}{x_{opt}} = 1 + \varepsilon = \rho$ (approximation ratio)

Orion

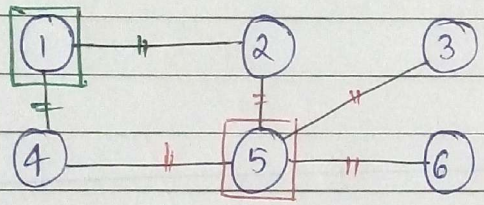
PAGE:
DATE: / /



$n = |U| = 2+2^2+2^3+\ldots+2^k = 2(2^k-1)$

$k \doteq O(\log n)$ $\Rightarrow$ $f = \frac{k}{2} = O(\log n)$

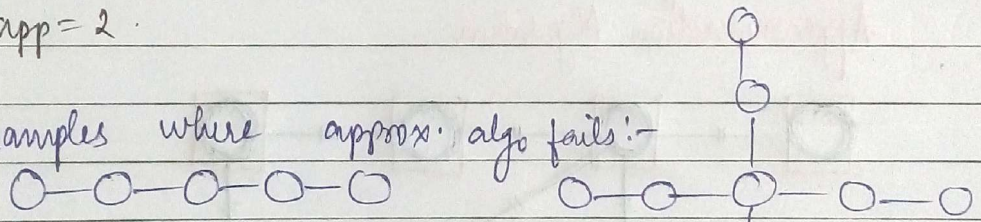
Vertex Cover Problem

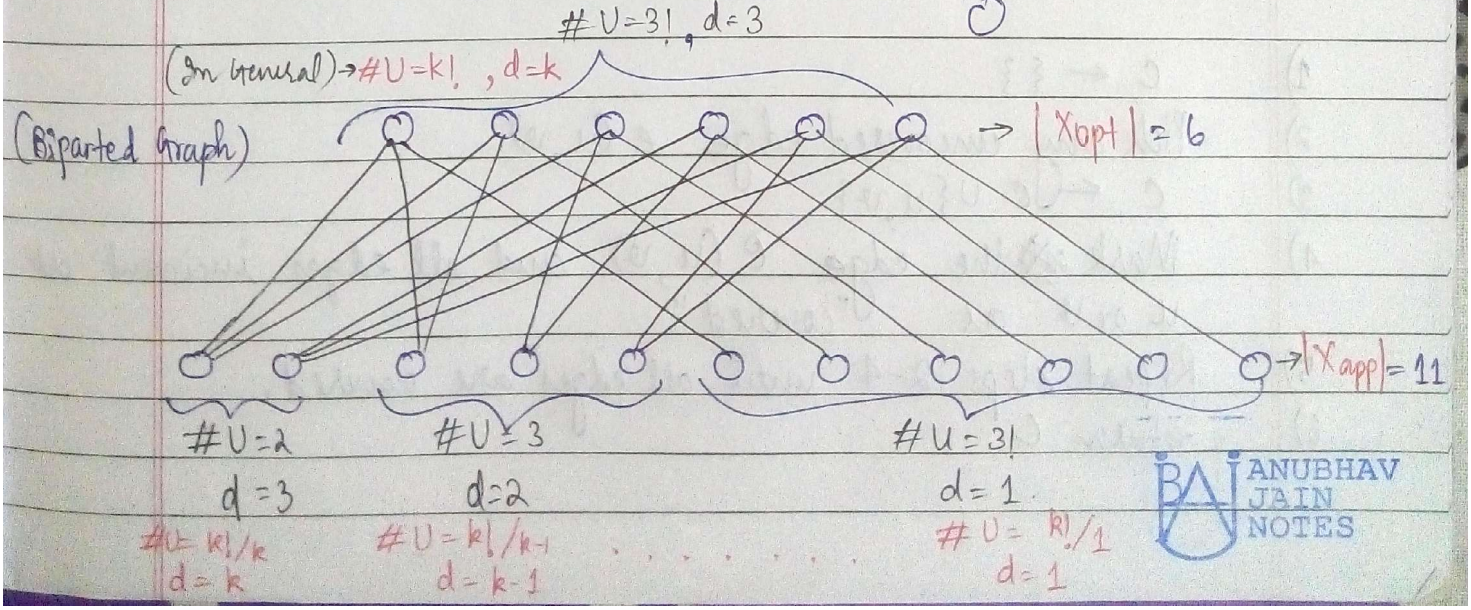


Choose vertex with max. degree.

$x_{app} = 2$.

Examples where approx. algo fails:-





ANUBHAV JAIN NOTES

Orion
PAGE:
DATE: / /

In general.

$|X_{opt}| = k!$

$|X_{approx}| = \frac{k!}{1} + \frac{k!}{2} + \frac{k!}{3} + \ldots + \frac{k!}{k} = O(\log k) \cdot k!$

$n = k! + O(\log k)\, k! = O(\log k)\, k!$

$\Rightarrow \log n = \log [O(\log k) + k!] = \log \log k + k \log k$

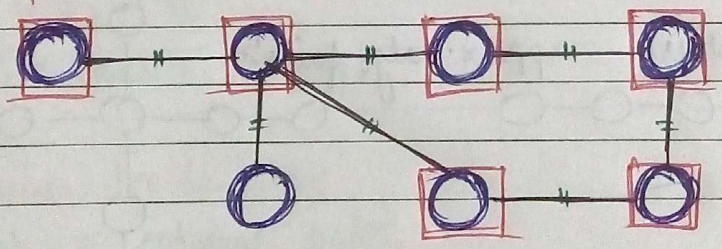
$\Rightarrow \log n = O(k \log k)$

$\rho = O(\log k)$

not a constant.

Vertex Cover Problem

Approximation Algorithm



1) $C \leftarrow \{\}$
2) Pick any uncovered edge $e(u, v)$
3) $C \leftarrow C \cup \{u, v\}$
4) Mark the edge $e(u, v)$ and all edges incident at u or v as "covered".
5) Repeat steps 2-4 until all edges are covered.
6) return C

Orion
PAGE:
DATE: / /

Approximation ratio (ρ)

In the optimal solution, for each edge there is at least one vertex. In the approx solution, for each edge there are at most two vertices. Hence $\boxed{\rho = 2}$

worst case

$2 |C_{opt}| \geq |C_{app}|$

13/11/17

Vertex Cover

Approximation Algorithm

$\rho = 2$ – Proof:-

C_{opt} = optimal cover
C_{app} = approximate cover
$e \in E$ is picked up arbitrarily and both the vertices of e are put in C_{app}

At least one of the two vertices of e must belong to C_{opt}. Because the edges selected for C_{app} do not share a vertex, when taken in pair.

$$\Rightarrow \frac{|C_{opt}|}{|C_{app}|} \geq \frac{1}{2}$$

$$\text{or, } \frac{|C_{app}|}{|C_{opt}|} = \rho \leq 2$$

Approximation Algorithm for TSP

Constant approximation ratio is not possible unless $P = NP$.

Proof:- Let $G(V, E)$ be an instance of Hamiltonian cycle Problem (HCP).

Orion

PAGE:
DATE: / /

Let $G'(V, E', w)$ be an instance of TSP such that G has a Ham-cycle if and only if G' has a cycle of total weight ??

$E' = V \times V$

$$w(e) = \begin{cases} 1 & \text{if } e \in E \\ \alpha & \text{if } e \notin E \end{cases}$$

$\forall\ e \in E'$

x_{opt} = min tour cost in G_1

x_{approx} = approximate tour cost in G_1

Then, $\frac{x_{app}}{x_{opt}} \leq f$

Let there be k edges in the optimal solution of TSP, which do not belong to E. So, there are $|V|-k$ edges that belong to E.

$\Rightarrow x_{opt} = k\alpha + (|V| - k)$

$\Rightarrow x_{app} \leq f \cdot x_{opt} = fk\alpha - fk + f|V|$

$= fk(\alpha - 1) + f|V|$

NO soln of HCP in $G \Leftrightarrow k \geq 1$

($k = 0 \Leftrightarrow$ Ham cycle exists in G)

$\Leftrightarrow \boxed{x_{app} \leq f|V|}$

$\Leftrightarrow x_{opt} \geq (\alpha - 1) + |V|$

$x_{opt} \leq x_{app} \leq f \cdot x_{opt}$

Put $\alpha = f|V| + 1$

$\Rightarrow x_{opt} \geq f|V| + |V|$

$\Rightarrow x_{opt} > f|V| \Rightarrow \boxed{x_{app} > f|V|}$

EndSem - 30% PreMidSem

Orion

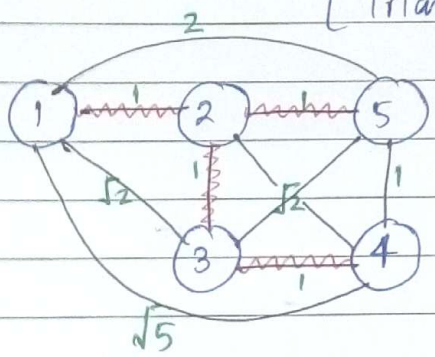
PAGE:
DATE: 14/11/17

Travelling Salesman Problem (TSP)

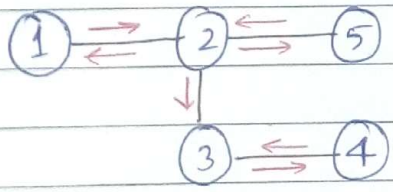
Euclidean TSP

$G = (V, E)$, where $E = V \times V$
with a weight function w

For any three vertices x, y, z in V,
$w(x,y) + w(y,z) \geq w(x,z)$ [Triangle Inequality]



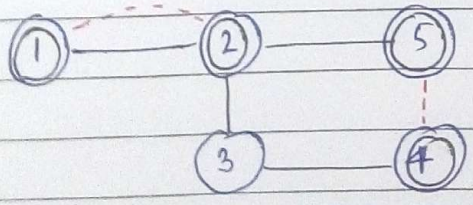
MST = {(1,2), (2,3), (3,4), (2,5)}.



Visit order: 1-2-3-4-3-2-5-1
⇓ 2-1
Capp: 1-2-3-4-5-1

Obs:

1) $w(MST) < w(Copt)$
2) $w(Capp) \leq 2 \cdot w(MST)$
3) $2w(MST) < 2w(Copt)$
$w(capp) < 2w(Copt) \Rightarrow$ **f ≤ 2**



1) Matching with min cost for vertices with odd degree.
2) Eulerian tour (every edge included exactly once)
→ Capp (1-2-5-4-3-1)