Module - 5

→ Approxmation Algorithm

- Approxmation Algorithm is a way of dealing with NP-completeness for optimization problem

→ It doesnot gurantee the best soln

→ An algorithm that returns near optimal soln is called Approxmation algorithm

→ **Goal**: It gives as close as possible to the optimum value in a reasonable amount of time which is polynomial time

→ Approxmation Algorithm are used when we donot have deterministic soln of the problem

→ Some Approxmation Algorithm are

1. finding vertex cover of a graph
2. Travelling salesperson problem
3. Minimum spanning tree

Scanned with CamScanner

Vertex cover problem

(Node cover problem)

Vertex cover means subset of vertices which cover every edge

-) A edge is covered if one of its end point choosen.

Idea: Finding a vertex that covers all the edges.

Algorithm:

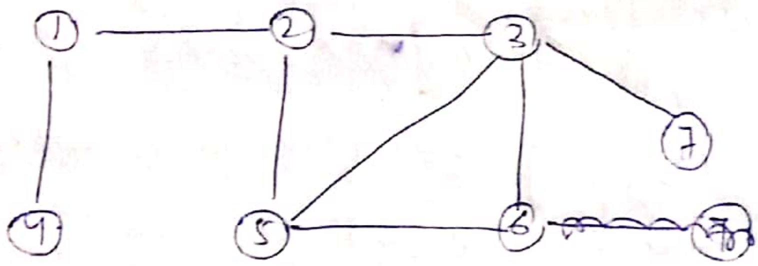
Step-1: Finding vertex with maximum degree

Step-2: Add v to the solution and remove all its incident edge from the graph

Step-3: Repeat while all edge are covered

~~Step-4~~: this algorithm focus on finding a minimum single node cover a graph.

Ex:



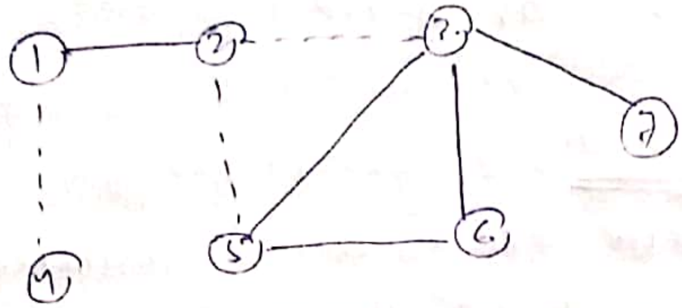
Iteration-1

choose edge (1,2)

C = {1, 2}

Remove edge

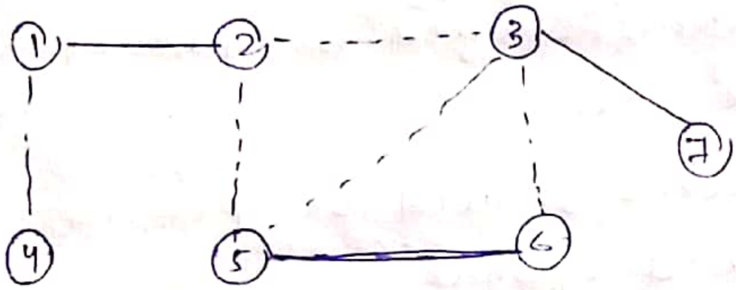
(1,4), (2,3) and (2,5)



Iteration-2 : choose {5, 6}

Now C = {1, 2, 5, 6}

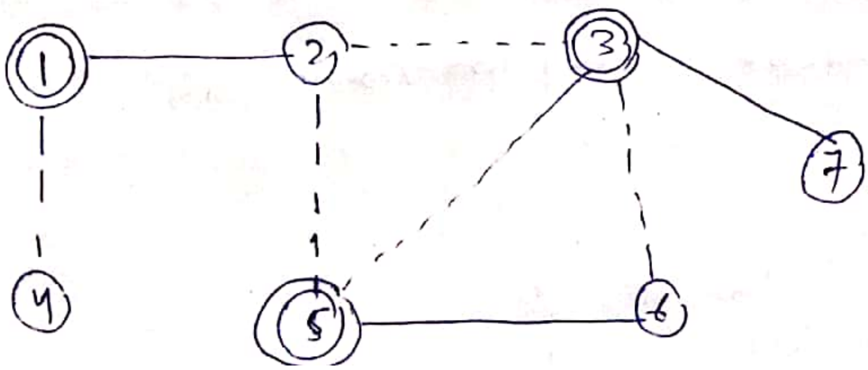
Remove (2,5) and {3,5}



Iteration-3 : choose edge {3,7}

C = {1, 2, 5, 6, 3, 7}

Remove edge is {3, 6}



Scanned with CamScanner

approximation vertex cover

$C = \{1, 2, 5, 6, 3, 7\}$ of size 7

The optimal vertex cover is

$C^* = \{1, 3, 5\}$ of size 3.

N-Queen problem:

The concept of N-Queen problem is that we need to place N-Queen in NXN matrix where in which the following condition should apply

→ No ~~row~~ two queens should be placed in the same row.

→ No two queens should be placed in the same column.

→ No two queens should be placed on the same diagonal.

4-Queen problem:

Here we have 4 queens

Q_1 should be placed in row1/col1

Q_2 should be placed in row2/col2

Q_3 should be placed in row3/col3.

Q_4 should be placed in row4/col4.

Scanned with CamScanner

(1) Consider the matrix and placed Q_1 in row1/col1.

	c_1	c_2	c_3	c_4
r_1	Q_1			
r_2				
r_3				
r_4				

(2) Now Q_2 should be placed in row2/col2 but it will crossed bcoz they are in the same diagonal. therefore Q_2 should be placed in $c_3 \times r_2$

	c_1	c_2	c_3	c_4
r_1	Q_1			
r_2			Q_2	
r_3				
r_4				

(3) Now Q_3 in row3/col3 but the Queen meet with each other bcoz they are on the same column. So there is no soln to place Q_3 in rows3/cols3. So backtrack again and move Q_1 to $c_3 \times r_1$ and also move Q_2 so that two queen should not attect the Q_3

place Q_1 in $r_1 \times c_3$

Q_2 in $r_2 \times c_1$

	c_1	c_2	c_3	c_4
r_1			Q_1	
r_2	Q_2			
r_3				
r_4				

(4) NOW place Q_3 on rows3/col3 but it cannot be placed bcoz the queen meet each other. So place the queen Q_3 on $r_3 \times c_4$

		Q_1	
Q_2			
			Q_3

NOW place Q_4 on $r_4 \times c_2$. so that no queen will meet with each other

	c_1	c_2	c_3	c_4
r_1			Q_1	
r_2	Q_3			
r_3				Q_3
r_4		Q_4		

NOW we got the soln on which we place the entire queen on 4x4 matrix where on which no two queen meet each other

i.e <3, 1, 4, 2>
or <2, 4, 1, 3>

Scanned with CamScanner

8 - queen problem

	1	2	3	4	5	6	7	8
1				Q_1				
2						Q_2		
3								Q_3
4		Q_4						
5							Q_5	
6	Q_6							
7			Q_7					
8					Q_8			

Travelling salesperson problem
(using Approximation algorithm)

In the travelling salesman problem introduce a complete undirected graph $G = \langle V, E \rangle$ that has a non negative integer cost $c(u,v)$ associated with each edge $(u,v) \in E$ and we must find a tour of G. with ~~minimum~~ minimal cost

$C(A)$ denotes the total cost of the edge in the subset $A \subseteq E$

$$\boxed{C(A) = \sum_{(u,v) \in A} c(u,v)}$$

Scanned with CamScanner

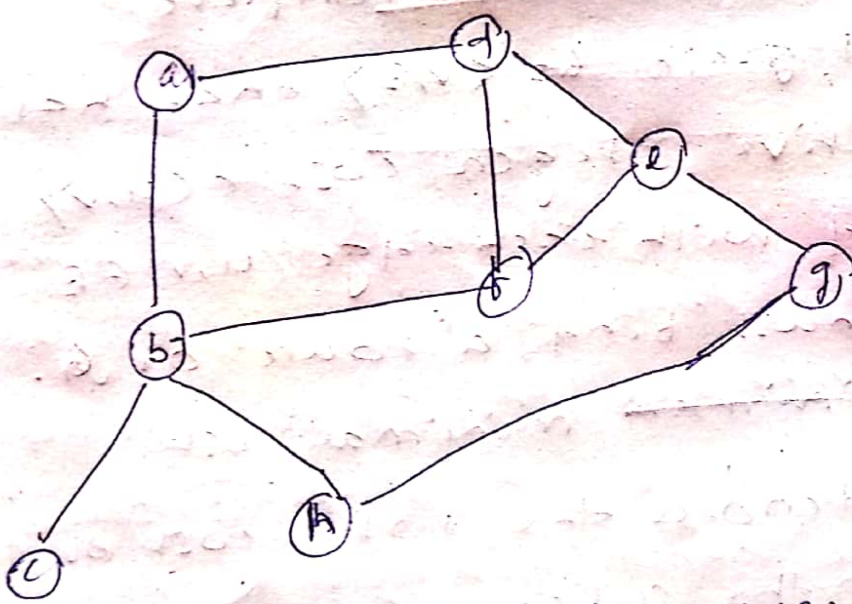
Hence the cost function c satisfies the ~~...~~ triangle inequality ~~...~~

if all vertices $u, v, w \in V$

$$\boxed{c(u,w) \leq c(u,v) + c(v,w)}$$

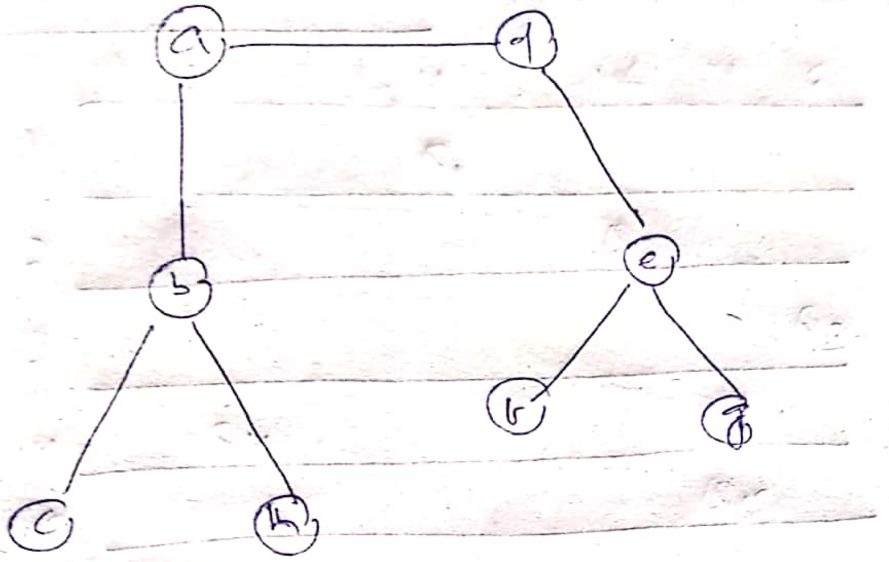
Approx–TSP (G(V,E))

1. compute a MST ~~of~~ T of G.
2. select any vertex r be the root of the tree
3. Let L be the list of vertices visited in a preorder tree walk of T.
4. Return the cycle that visits the vertices in the order.

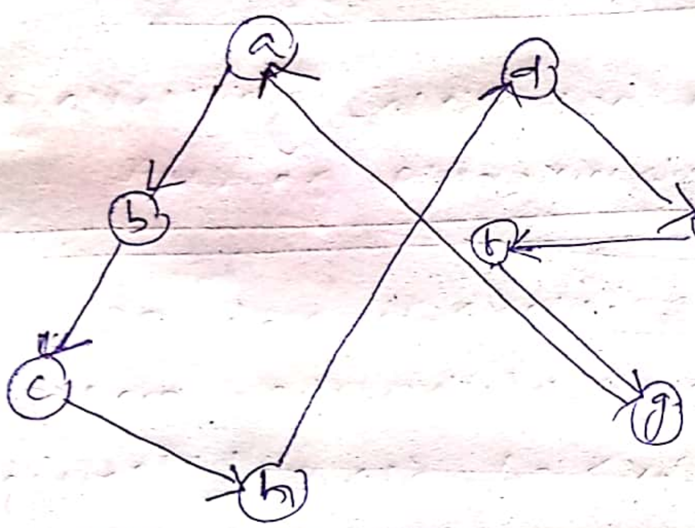


(Given set of points)

Scanned with CamScanner



preorder of tree(T) = abchdefg.



A preorder sequence gives a tour H.
Here saleman need to visit n cities
He want to start a tour at a city.
and visit every cities exactly once
and finish the tour at the city from
where he starts.

Scanned with CamScanner