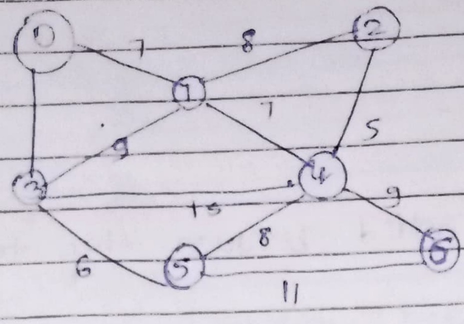
Name - Tanvi Kharade [INFT A]
22101A0034

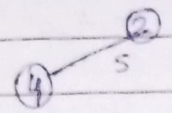
Adsa

Assignment no.5

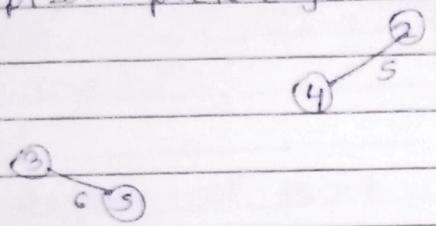
Q.1) K-step



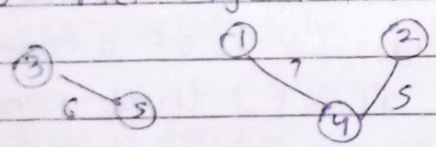
→ Step(1) :- Pick edge 2-4, No cycle formed.



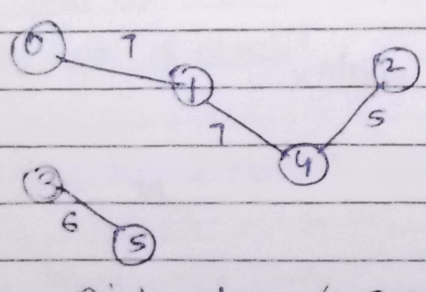
Step(2) :- Pick edge 3-5. No cycle formed.



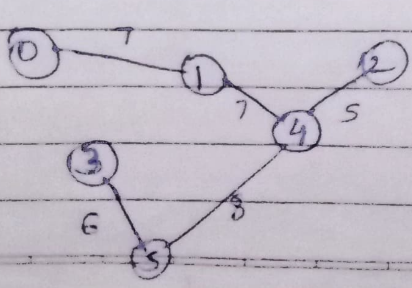
Step(3) :- Pick edge 1-4. No cycle formed



Step 4 :- Pick edge 0-1. No cycle formed

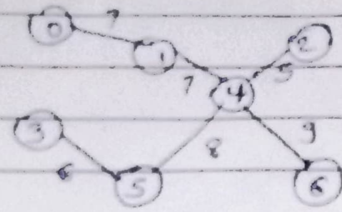


Step 5 :- Pick edge 4-5 No cycle formed



SDS Page No. Date

step 6:- Pick edge 4-6 No-cycle formed



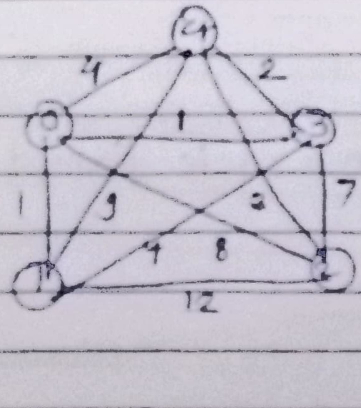
∴ Rest of edge cannot added because they formed cycle

∴ Minimum Spanning Tree Cost = 5 + 6 + 7 + 7 + 8 + 9
= 42

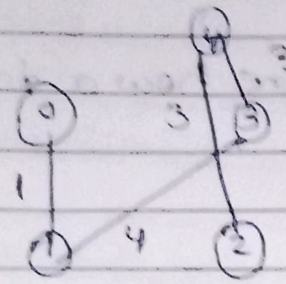
Q.2) ⇒ $W = \begin{bmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{bmatrix}$

1) A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges (v-1) of a connected edge-weighted undirected graph G(v, E) that connects all the vertices together without any cycles and with minimum possible total edge weight

Graph for Adjacency matrix



Minimum possible weight of a spanning tree taking the ~~t~~ vertex 0 as a leaf node.



Sum of the weight of the minimum spanning Tree = 1+4+2+3 = 10

∴ So, 10 is the minimum possible weight of a spanning tree when taking vertex 0 as a leaf node.

Q.3)

feasible sol^n	Optimal sol^n
(i) A feasible sol^n is one that satisfies all the problems constraints	(i) Optimal sol^n is the best possible sol^n among all feasible sol^n.
(ii) In context, a feasible sol^n may not necessarily be the best or most efficient, but it is valid	(ii) It minimizes the objective function while still satisfying constraints
iii) For ex:- In knapsack problem, any combination of items that does not exceed the knapsack's weight limit of feasible sol^n.	iii) In the knapsack problem, the optimal sol^n maximizes the total value of items in the knapsack without exceed the weight limit.

$n = 7, m = 20$

$P\{1:7\} = \{8, 5, 6, 7, 6, 12, 3\}$

$w\{1:7\} = \{2, 10, 8, 7, 6, 4, 11\}$

Object	P	w	P/w
1	8	2	4
2	5	10	0.5
3	6	8	0.75
4	7	7	1
5	6	6	1
6	12	4	3
7	3	11	0.27

Descending order

Object	P	W	P/W
1	8	2	4
6	12	4	3
4	7	7	1
5	6	6	1
3	6	8	0.75
2	5	10	0.5
7	3	11	0.27

$= \left(2+4+7+6+\left(8\times\frac{1}{8}\right)\right)$

$= \left(8+12+7+6+\left(6\times\frac{1}{8}\right)\right)$

$= 33.75$

$= \{1, 0, \left(1\times\frac{1}{8}\right), 1, 1, 1, 0\}$

Dijk

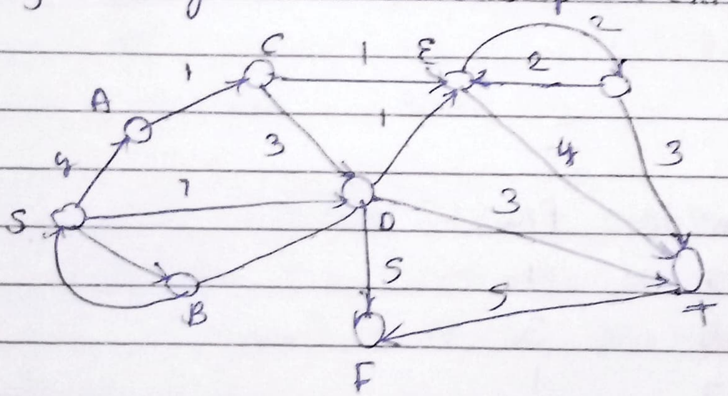
Q.4) → i) The constraints must be followed by Dijkstra's algo to solve the single source shortest path using greedy method as follow:-

i) Non-negative weights:- All edges must have non-negative weights.

ii) Connected Graph:- The graph should be connected for the algorithm to reach all vertices.

iii) Greedy Approach:- The algorithm selects the vertex with smallest tentative distance at each step.

iv) Relaxation:- Update the shortest path to each vertex by checking if a shorter path exist through another vertex.



∴ Given graph vertices:- S, A, B, C, D, E, F, G, T

Selected vertex	S	A	B	C	D	E	F	G	T
	0	∞	∞	∞	∞	∞	∞	∞	∞
S		4	3	7	∞	∞	∞	∞	∞
B		4	3	7	8	9	∞	∞	∞
A		4	3	5	8	∞	∞	∞	∞
C		4	3	5	8	6	∞	∞	9
E		4	3	5	8	6	∞	8	10

D	4	3	5	8	6	13	8	10
G	4	3	5	8	6	13	8	10
T	4	3	5	8	6	13	8	10
F	4	3	5	8	6	13	8	10

∴ S - B - A - C - E - D - G - T - F

∴ 0 - 3 - 4 - 5 - 6 - 8 - 8 - 10 - 13 //

∴ This gives shortest ~~an~~ path and distance from vertex S of source as to all vertex of remaining using Dijkstra's algorithm.

Q. 5) 1) The Greedy method follows a problem-solving 'strategy where at each step, the best possible choice (according to certain criterion) is made with hope that these local optimal sol^n.

The general step involved in Greedy method.

i) Formula the problem:- Define problem in terms of set of choice constraints, and an objective function.

ii) Determine feasibility:- Identify the choice that satisfy the constraints of problem.

iii) Make a Greedy Choice:- Select the choice that looks best according to the objective function. This choice is typically one that maximize or minimize the objective function. This choice is typically one that maximize or minimize the objective function.

SDS Page No. Date

iv) Check for Optimaly :-
verify if the current choice leads to best solⁿ among all possible solutions
v) Iterate :- Repeate process until all choice are made or until no further improvement can be made.

n = 7
I {1:7} = {1, 2, 3, 4, 5, 6, 7}
P {1:7} = {3, 5, 20, 18, 1, 6, 30}
D {1:7} = {1, 3, 4, 3, 2, 1, 2}
As maximum waiting time is not given in given problem
Take maximum waiting time that is Job which has more deadline i.e. n = 4

Job	7	3	4	6	2	1	5
Profit	30	20	18	6	5	3	1
Deadline	2	4	3	1	3	1	2

0	1	2	3	4
6	7	4	3	
6	30	18	20	
1	2	3	4	

∴ maximum Profit = {~~5,~~ 6 + 30 + 18 + 20}
= 74
= {0, 0, 1, 1, 0, 1, 1}