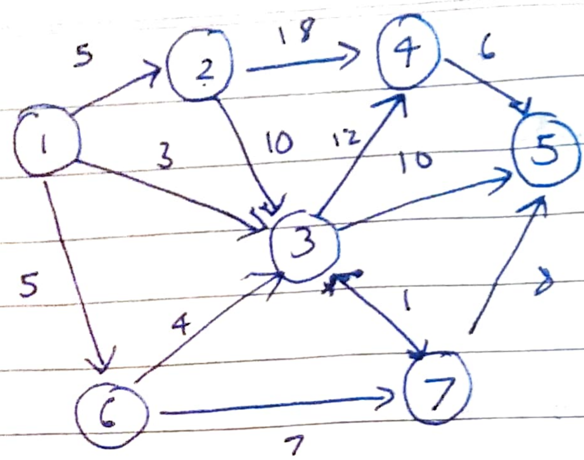
1



Total nodes = 7

$\therefore$ Total iteration = 7 - 1

= 6

Edges = (1 → 2) (1 → 6) (1 → 3) (2 → 3) (3 → 4) (2 → 4) (3 → 5)
(4 → 5) (6 → 3) (3 → 7) (6 → 7) (7 → 5)

Initially we will start from node 1, Node weight of rest will be ∞ as they aren't explored

1st iteration

Edge : (1 → 2)

Weight of 1 = 0

Cost of (1, 2) = 5

$\therefore$ 1 + c(1, 2) = 0 + 5

= 5

∞ > 5

$\therefore$ d[2] = 5

Edge : (1 → 6)

d[1] + c(1, 6) = 5

$\therefore$ d(6) = 5

Edge $1 \to 3$

$d[1] + c[1,3] = 3$

$\therefore d[3] = 3$

Edge $2 \to 3$

$d[2] + c[2,3] = 15$

$d[3] < d[2] + c[2,3]$

$\therefore d[3] = 3$

Edge $3 \to 4$

$d[3] + c[3,4] = 15$

$\therefore d[4] = 15$

Edge :- $2 \to 4$

$d[2] + c(2,4) = 23$

$d[4] < d[2] + c(2,4)$

$\therefore d[4] = 15$

Edge :- $3 \to 5$

$d[3] + c[3,5] = 13$

$d[5] = 13$

Edge :- $4 \to 5$

$d[4] + c(4,5] = 21$

$d[5] < d[4] + c(4,5)$

$\therefore d[5] = 13$

Edge :- $[6 \to 3]$

$d[6] + c(6,3] = 9$

$d[3] < d[6] + c(6,3)$

$\therefore d[3]\ 3$

Edge :- $[3 \to 7]$

$d[3] + c(3,7) = 4$

$d[7] = 4$

Edge : $[6 \to 7]$

$d[6] + c(6,7) = 12$

$\therefore d[7] < d[6] + c(6,7)$

$\therefore d[7] = 4$

Edge :- $[7 \to 5]$

$d[7] + c(7,5) = 12$

$d[5] > d[7] + c(7,5)$

$\therefore d[5] = 12$

2^{nd} iteration

Edge :- $(1 \to 2)$

$d[1] + c(1,2] = 5$

$\therefore d[2] = 5$

Edge :- $(1 \to 6)$

$d[1] + c(1,6) = 5$

$d[6] = 5$

Edge :- $(1 \to 3)$

$d[1] + c(1,3) = 3$

$\therefore d[3] = 3$

Edge $(2 \to 3)$
$d[2] + c[2,3] = 15$
$d[3] = 3$

Edge:- $(3 \to 4)$
$d[3] + c(3,4) = 15$
$d[4] = 15$

Edge $(2 \to 4)$
$d[2] + c(2,4) = 23$
$\therefore d[4] = 15$

Edge. $(3 \to 5)$
$d[3] + c(3,5) = 13$
$d[5] = 12$

Edge:- $(4 \to 5)$
$d[4] + c(4,5) = 21$
$d[5] = 12$

Edge:- $[6 \to 3]$
$= d(6) + c(6,3) = 9$
$d(3) = 3$

Edge:- $(3 \to 7)$
$d(3) + c(3,7) = 4$
$d(7) = 4$

Edge $(6 \to 7)$
$= d(6) + c(6,7) = 12$
$d(7) = 4$

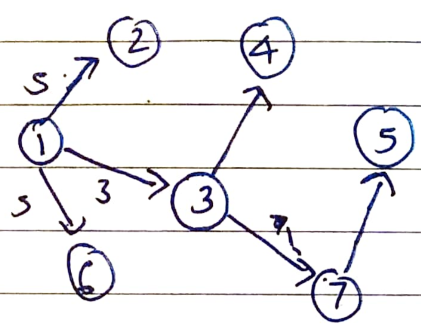
Edge $(7 \to 5)$

$d(7) + c(7,5) = 12$

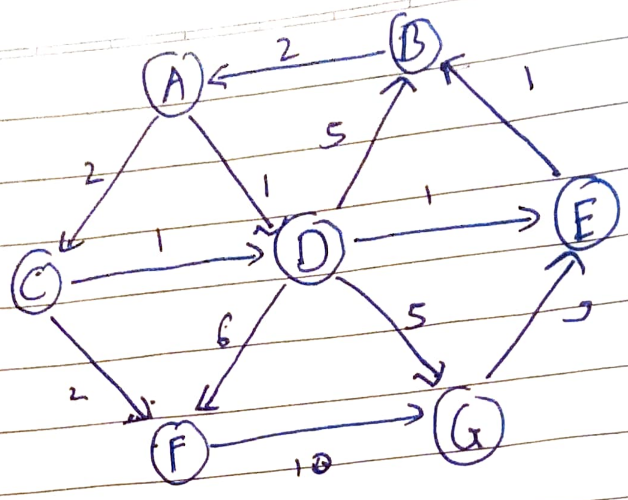
$d(5) = 12$

As this iteration has no change, edge distance sum will be same for rest.

$\therefore$ Final Graph



2.



Source(A)

Explored	Unexplored
	$\overset{0}{A}$ $\overset{\infty}{B}$ $\overset{\infty}{C}$ $\overset{\infty}{D}$ $\overset{\infty}{E}$ $\overset{\infty}{F}$ $\overset{\infty}{G}$
$\overset{0}{A}$	$\overset{\infty}{B}$ $\overset{2A}{C}$ $\overset{1A}{D}$ $\overset{\infty}{E}$ $\overset{\infty}{F}$ $\overset{\infty}{G}$
$\overset{0}{A}$ $\overset{1A}{D}$	$\overset{6D}{B}$ $\overset{2A}{C}$ $\overset{2D}{E}$ $\overset{7D}{F}$ $\overset{6D}{G}$
$\overset{0}{A}$ $\overset{1A}{D}$ $\overset{2A}{C}$	$\overset{6D}{B}$ $\overset{2D}{E}$ $\overset{4C}{F}$ $\overset{6D}{G}$
A $\overset{1A}{D}$ $\overset{2A}{C}$ $\overset{2D}{E}$	$\overset{3E}{B}$ $\overset{4C}{F}$ $\overset{6D}{G}$
A $\overset{1A}{D}$ $\overset{2A}{C}$ $\overset{2D}{E}$ $\overset{3E}{B}$	$\overset{4C}{F}$ $\overset{6D}{G}$
A $\overset{1A}{D}$ $\overset{2A}{C}$ $\overset{2D}{E}$ $\overset{3E}{B}$ $\overset{4C}{F}$	$\overset{6D}{G}$

$\overset{0}{A}$ $\overset{1A}{D}$ $\overset{2A}{C}$ $\overset{2D}{E}$ $\overset{3E}{B}$ $\overset{4C}{F}$ $\overset{6D}{G}$

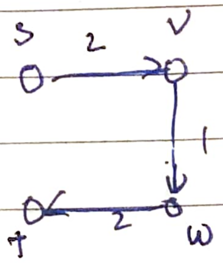
Vertex	Known ?	Cost	Path
A		0	Source
B		3	A → D → E → B
C		2	A → C
D		1	A → D
E		2	A → D → E
F		4	A → C → F
G		6	A → D → G

PAGE No
DATE

3. Following statement is False

Counter:-

Consider a graph with vertices s, v, w, t and edges (s, v), (v, w), (w, t) and capacities of 2 on (s, v) & (w, t) and capacity of 1 on (v, w)



Then the maximum flow has value 1 & this does not saturate the edge out of 's'

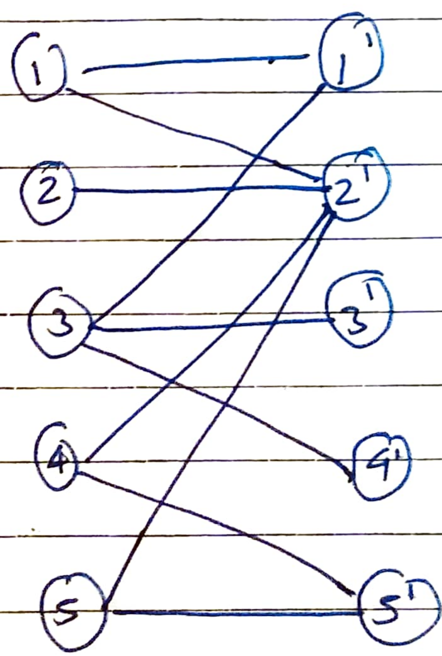
PAGE No.

DATE / /

4 Let S be a subset of nodes and let N(s) be the set of nodes adjacent to nodes in S

Observation :- If a bipartite graph G = L U R , E has a perfect matching , then |N(s)| ≥ |s| for all the subset S⊆L

Proof :- Each node in S has to be matched to a different nod in N(s)



PAGE No
DATE / /

5. a) iv) There exist exactly one edge which can be removed to disconnect the graph into 2 pieces, one containing s & other containing t

Proof :- If the final flow is 1 then this means that the minimum cut has only one edge of weight 1 connecting u & v. There is no other edges connecting u in S & other vertex T otherwise the flow would be more than 1

b) Answer is False

This algo has one for loop, Hence if we consider x as n then loop will iterate n/2 time so
∴ ~~TI~~ ~~Ir~~ Time complexity = $O(n/2)$

∴ Complexity = linear

∴ Its ~~or~~ not polynomial