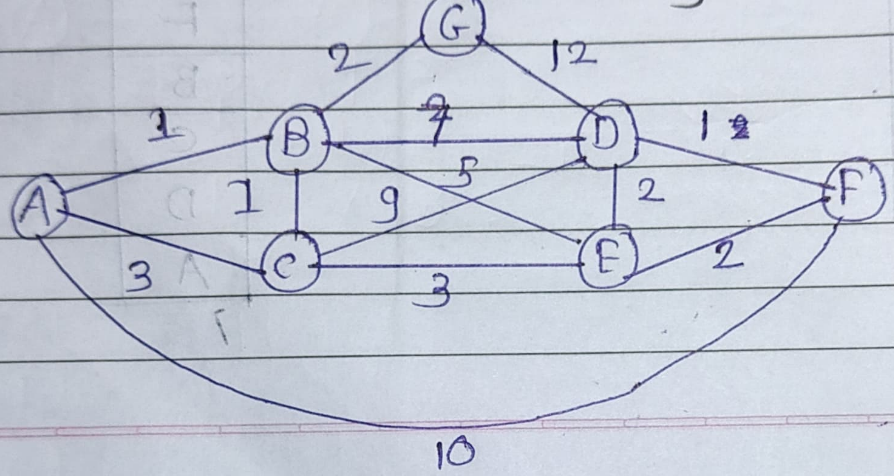
Q2 Explain Single Source shortest path algorithm and apply it on following given graph. Calculate the single-source shortest path from A to every other vertex.



Single source shortest path algorithm

The single source shortest path algorithm is a type of graph traversal algorithm that is used to find the shortest path between source node and all other nodes in a weighted graph. Algorithm compute the shortest path from starting vertex.

if $[d[u] + c(u,v) \leq d[v]]$

$$d[v] = d[u] + c[u,v]$$

Relaxation of edges $= |V| - 1$

Time Complexity

1) Only one node is connected to all other node

$= O$ [E. No. of Relaxation]

$= O\,[n.n]$

$= O(n^2)$

2) Each node is connected to every other node (i.e Graph is complete graph)

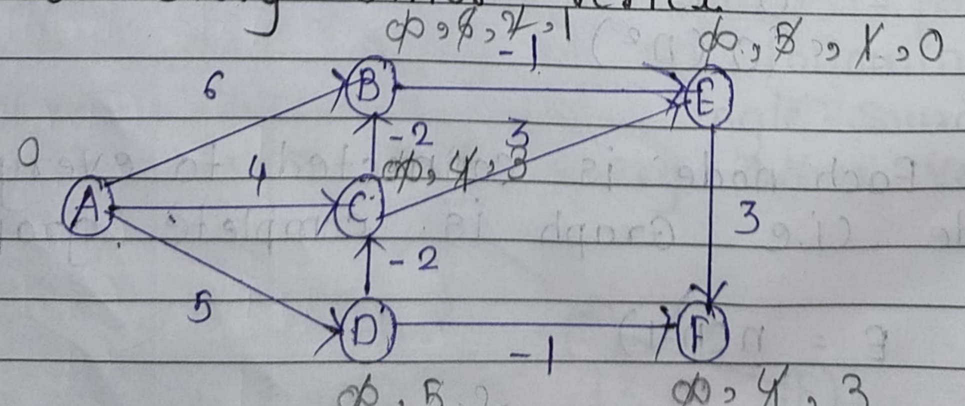
$$E = \frac{n(n-1)}{2}$$

$$= O\left[n.\ \frac{n(n-1)}{2}\right] = O(n^3)$$

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
G	1	10	8	6	10	3
E	1	10	8	6	10	3
D	1	9	8	6	8	3
F	1	9	8	6	8	3
C	1	9	8	6	8	3

v	d[v]
B	1
C	9
D	8
E	6
F	8
G	3

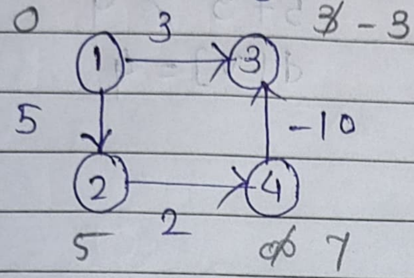
Q3 Write limitations of single sources shortest path algorithm and apply bellman ford algorithm on following given graph. calculate shortest path from A to every other vertex.



→ limitation of single source shortest path algorithm

It can not handle negative ~~values~~ edges. This leads to acyclic graph and can-not obtain right shortest path. It's greedy

EX



→ Relaxation $= |V| - 1$

$= 6 - 1 = 5$

(u, v)

if $(d[u] + c(u,v) < d[v])$

$d[v] = d[u] + c[u,v]$

Edges = (A, B) (A, C) (A, D) (B, E) (C, B) (C, E) (D, C) (D, F) (E F)

① First step

1) (A, B)

$0 + 6 < \infty$

$d[v] = 6$

2) (A, C)

$0 + 4 < \infty$

$d[v] = 4$

3) (A, D)

$0 + 5 < \infty$

$d[v] = 5$

4) (B, E)

$6 + (-1) < \infty$

$5 < \infty = d[v] = 5$

5) (C, B)

$4 + (-2) < \infty$

$2 < \infty = d[v]\ 2$

6) ~~(C, E)~~

~~4+(-3) < ∞~~

~~1 < ∞ = d[v] = 1~~

6> (C, E)

$4 + 3 \not< 5$

$d[V] = 5$

7> (D, C)

$5 + (-2) < 4$

$3 < 4$

$d[V] = 3$

8> (D, F)

$5 + (-1) < \infty$

$4 < \infty$

$d[V] = 4$

9> (E, F)

$5 + 3 > 4$

$d[V] = 4$

(2) Second step

(A, B) = 2   (A, C) = 3   (A, D) = 5

(B, E) = $2 + (-1) < 5$
= $1 < 5$
= $d[V] = 1$

(C, B) = $3 + (-2) < 2$
= $1 < 2$
= $d[V] = 1$

(C, E) = 1   (D, C) = 3   (D, F) = 4

(E, F) = 4

(3) Third step

(A, B) = 1
(A, C) = 3
(A, D) = 5
(B, E) = $1 + (-1) = 0$
(C, B) = 1
(C, E) = 0
(D, C) = 3
(D, F) = 4
(E, F) = $0 + 3 = 3$ $d[V] = 3$

Page No.
Date

(4) Fourth step

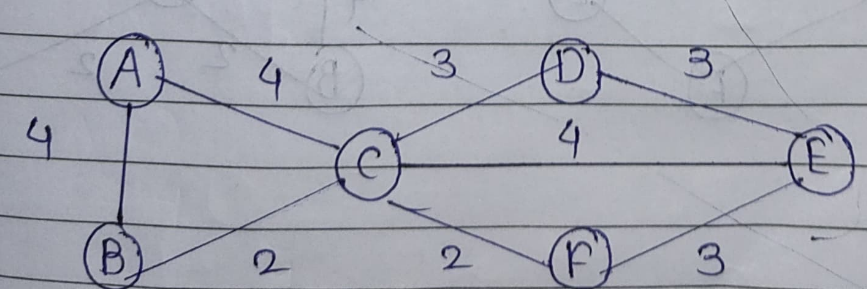
(A,B) = 1
(A,C) = 3
(A,D) = 5
(B,E) = 0
(C,B) = 1
(C,E) = 0
(D,C) = 3
(D,F) = 3
(E,F) = 3

(5) fifth step

(A,B) = 1
(A,C) = 3
(A,D) = 5
(B,E) = 0
(C,B) = 1
(C,E) = 0
(D,C) = 3
(D,F) = 3
(E,F) = 3

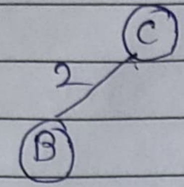
A	→ 0
B	→ 1
C	→ 3
D	→ 5
E	→ 0
F	→ 3

4) Differentiate between Prim's and kruskal's algorithm and apply both algorithm on following graph

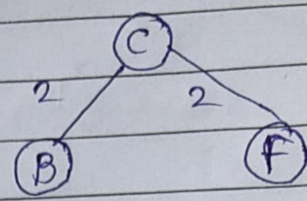


1) Prim's algorithm

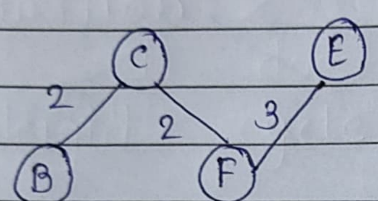
step 1



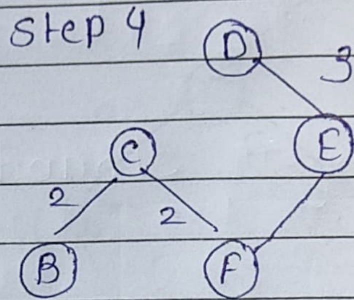
step 2



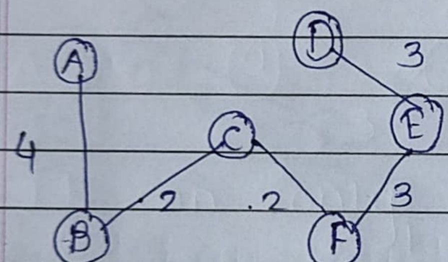
step 3



step 4



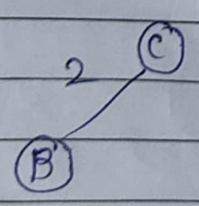
step 5



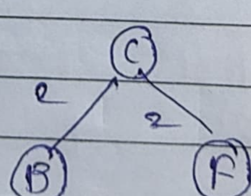
$$\text{cost} = 4+2+2+3+3$$
$$= 14$$

2) Kruskal's algorithm

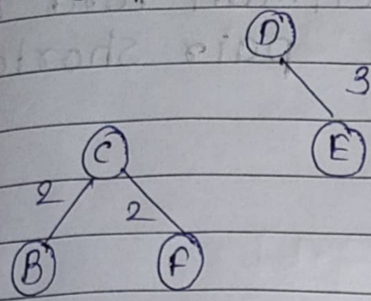
step 1



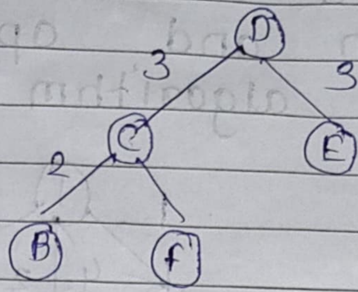
step 2



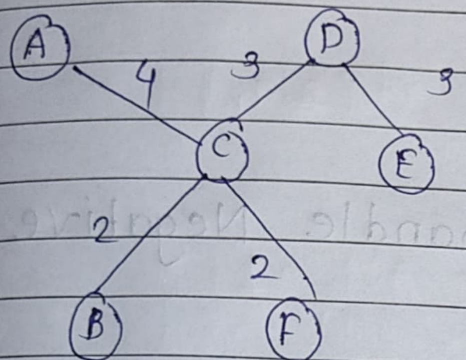
step 3



step 4



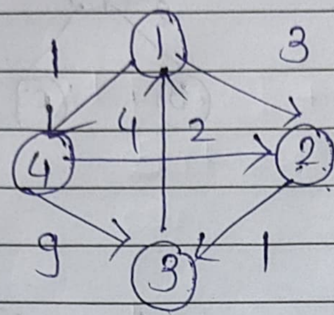
step 5



$$\text{cost} = 4+3+3+2+2$$
$$= 14$$

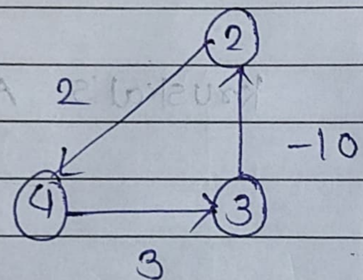
Prim's Algorithm	Kruskal's Algorithm

Q5 Write limitions of bellman ford algorithm and apply all pair shortest path algorithm.



→ limition

It does not handle Negative Weight cycle



$5+3+(-10) = -2$

It can only work for directed graph

→

$$A^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 2 & \infty & 0 & \infty \\ \infty & 2 & 9 & 0 \end{bmatrix} \end{array}$$

$$A^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 2 & 5 & 0 & 3 \\ \infty & 2 & 9 & 0 \end{bmatrix} \end{array}$$

$A'[2,3] = A^{\circ}[2,1] + A^{\circ}[1,3]$

$1 = \infty + \infty$

$1 < \infty$

$A'[2,4] = A^{\circ}[2,1] + A^{\circ}[1,4]$

$\infty = \infty + 1$

$A'[3,2] = A^{\circ}[3,1] + A^{\circ}[1,2]$

$\infty = 2 + 3$

$\infty > 5$

$A'[3,4] = A^{\circ}[3.1] + A^{\circ}[1,4]$

$\infty = 2 + 1$

$\infty > 3$

$A'[4,2] = A^{\circ}[4.1] + A^{\circ}[1,4]$

$2 = \infty + 1$

$2 < \infty.$

$A'[4.3] = A^{\circ}[4.1] + A^{\circ}[1.3]$

$9 = \infty + \infty$

$9 < \infty$

$$A^2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \underline{\underline{4}} & \underline{\underline{1}} \\ \infty & 0 & 1 & \infty \\ \underline{\underline{2}} & 5 & 0 & \underline{\underline{3}} \\ \underline{\underline{\infty}} & 2 & \underline{\underline{3}} & 0 \end{bmatrix} \end{array} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & 4 & 1 \\ \infty & 0 & 1 & \infty \\ 2 & 5 & 0 & 3 \\ \infty & 2 & 3 & 0 \end{bmatrix} \end{array}$$

$A^2[1,3] = A^1[1,2] + A^1[2,3]$

$\infty = 3 + 1$

$\infty > 4$

$A^2[1,4] = A^1[1,2] + A^1[2,4]$

$1 = 3 + \infty$

$1 < \infty$

$A^2[3,1] = A^1[3,2] + A^1[2,1]$

$2 = 5 + \infty$

$2 < \infty$

$A^2[3,4] = A^1[3,2] + A^1[2,4]$

$\cancel{\infty}\ 3 < 5 + \infty$

$A^2[4,1] = A^1[4,2] + A^1[2,1]$

$\infty = 2 + \infty$

$\infty = \infty$

$A^2[4,3] = A^1[4,2] + A^1[2,3]$

$9 = 2 + 1$

$9 > 3$

$$A^3 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 4 & 1 \\ 2 & 3 & 0 & 1 & 4 \\ 3 & 2 & 5 & 0 & 3 \\ 4 & 5 & 2 & 3 & 0 \end{array} = \begin{bmatrix} 0 & 3 & 4 & 1 \\ 3 & 0 & 1 & 4 \\ 2 & 5 & 0 & 3 \\ 5 & 2 & 3 & 0 \end{bmatrix}$$

$A^3[1,2] = A^2[1,3] + A^2[3,2]$

$3 = 4 + 5$

$3 < 9$

$A^3[1,4] = A^2[1,3] + A^2[3,4]$

$1 = 4 + 3$

$1 < 7$

$A^3[2,1] = A^2[2,3] + A^2[3,1]$

$\infty = 1 + 2$

$\infty > 3$

$A^3[2,4] = A^2[2,3] + A^3[3,4]$

$\infty > 1 + 3$

$\infty > 4$

$A^3[4,1] = A^2[4,3] + A^2[3,1]$

$\infty = 3 + 2$

$\infty > 5$

$A^3[4,2] = A^2[4,3] + A^2[3,2]$

$2 = 3 + 5$

$2 < 8$

$$A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 4 & 1 \\ 2 & 3 & 0 & 1 & 4 \\ 3 & 2 & 5 & 0 & 3 \\ 4 & 5 & 2 & 3 & 0 \end{array}$$

$A^4[1,2] = A^3[1,4] + A^3[4,2]$

$3 = 1 + 2$

$3 \nless 3$

$A^4[1,3] = A^3[1,4] + A^3[4,3]$

$4 = 1 + 3$

$4 = 4$

$A^4[2,1] = A^4[2,4] + A^4[4,1]$
$3 < 4 + 5$
$3 < 9$

$A^4[2,3] = A^4[2,4] + A^4[4,3]$
$1 < 4 + 3$
$1 < 7$

$A^4[3,1] = A^4[3,4] + A^4[4,1]$
$2 < 3 + 5$
$2 < 8$

$A^4[3,2] = A^4[3,4] + A^4[4,2]$
$5 = 3 + 2$
$5 = 5$

Q6 Differentiate DFS and BFS Algorithm

DFS	BFS
1) DFS stand for Depth First search	1) BFS stand for breadth First search
2) DFS uses a stack to find shortest path	2) BFS uses a queue to Find shortest path
3) Time complexity of DFS = O(V+E) V is Vertices E is Edges.	3) Time complexity of BFS = O(V+E) V is Vertices E is Edges.

DFS	BFS
4) DFS is implemented using LIFO principle	4) BFS is implemented uing FIFO principle.
5) DFS require less memory	5) BFS require more memory
6) DFS is a traversal approach in which the traverse begins at the root node and proceeds through the	6)

Step 1

	B	C	D	E	F	G
	1	3	∞	∞	10	∞

Step 2

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
	1	2	8	6	10	3

Step 3

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
C	1	2	8	6	10	3
	1	2	8	5	10	3

Step 4

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
C	1	2	8	6	10	3
G	1	2	8	5	10	3
	1	2	8	5	10	3

Step 5

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
C	1	2	8	6	10	3
G	1	2	8	5	10	3
E	1	2	8	5	10	3
	1	2	7	5	7	3

Step 6

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
C	1	2	8	6	10	3
G	1	2	8	5	10	3
E	1	2	8	5	10	3
D	1	2	7	5	7	3
	1	2	7	5	7	3

Step 7

	B	C	D	E	F	G
B	1	3	∞	∞	10	∞
C	1	2	8	6	10	3
G	1	2	8	5	10	3
E	1	2	8	5	10	3
D	1	2	7	5	7	3
F	1	2	7	5	7	3

V	d[V]
B	1
C	2
D	7
E	5
F	7
G	3

DFS	BFS
4) DFS is implemented using LIFO principle	4) BFS is implemented uing FIFO principle.
5) DFS require less memory	5) BFS require more memory
6) DFS is a traversal approach in which the traverse begins at the root node and proceeds through the nodes as far as possible until we reach the node with no unvisited nearby nodes	6) BFS is a traversal approach in which we first walk through all nodes on the same level before moving on to the next level
7) DFS builds the tree level ~~by~~ ~~level~~ sub-tree by subtree	7) BFS builds the tree level by level
8) Here ~~siblings~~ ~~are~~ visited ~~before~~ ~~the~~ children are visited before sibling	8) Here siblings are visited before the children.
9) DFS is fast a compared to BFS	9) BFS is slow as compared to DFS.