

## HW #4 Solutions

1. How many different spanning trees (that connect all nodes) can be formed on top of an  $n$ -node graph that has an edge between every pair of nodes. Explain how you reached your answer.

Answer:

*This question was withdrawn from this set.*

2. Let  $G$  be a graph where each edge has 2 weight functions,  $w$  and  $v$ , where  $v(e) = w(e) + 1$ ,  $w(e) > 0$  for each edge  $e \in G$ .

- (a) If tree  $T$  is a minimum spanning tree using weight function  $w$ , is it also a minimum spanning tree using weight function  $v$ ? Explain why this is true or give a counterexample.

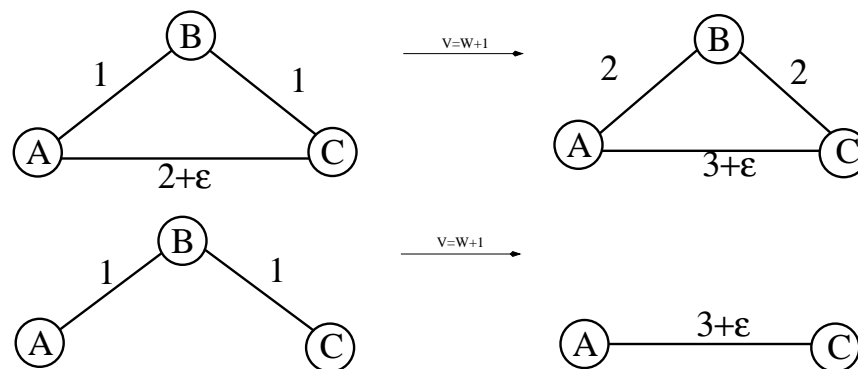
Answer:

Yes, If  $G$  has  $n$  nodes and  $T$  is a tree with weight  $W$  when using weight function  $w$ , then  $T$  is a tree with weight  $W + n - 1$  when using weight function  $v$ . Since the difference in weight of any tree using the two weight functions is fixed, it follows that any m.s.t.  $M$  with weight  $W_m$  using  $w$  has weight  $W_m + n - 1$  when using weight  $v$ , and any tree that is not a spanning tree with  $weight > W_m$  using  $w$  has  $weight > W_m + n - 1$  when using  $v$ .

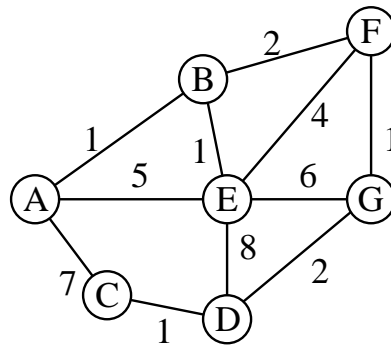
- (b) If tree  $T$  is a shortest path tree using weight function  $w$ , is it also a shortest path tree using weight function  $v$ ? Explain why this is true or give a counterexample.

Answer:

No, because the shortest path can contain arbitrary number of edges. Using this fact we can create counterexamples similar to the following one ( $\epsilon \ll 1$ ):



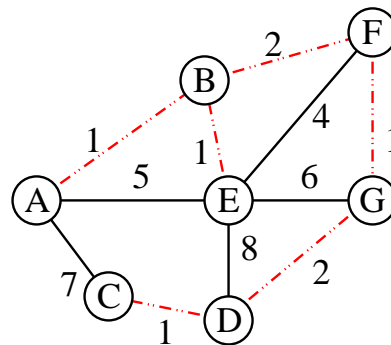
(for shortest paths from A to C)



3. Compute an MST for the above graph using either algorithm covered in class.

Answer:

An MST for the above graph is:



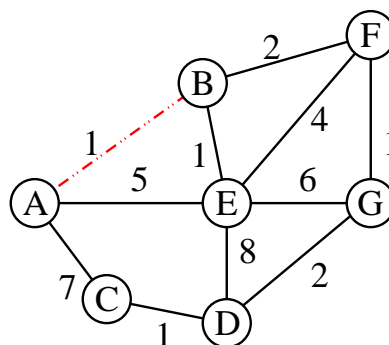
4. Compute a shortest path tree rooted at node A.

- (a) using Dijkstra's algorithm.

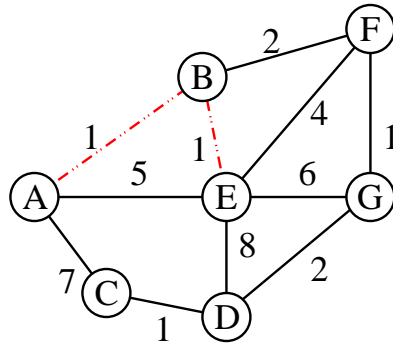
Answer:

The Shortest Path Tree for node A using Dijkstra's algorithm is:

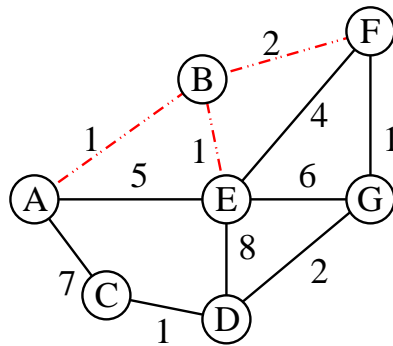
STEP 1 IN DIJKSTRA'S ALGORITHM



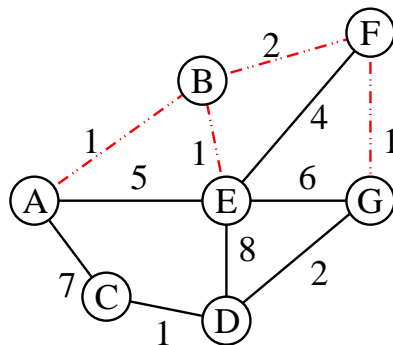
## STEP 2 IN DIJKSTRA's ALGORITHM



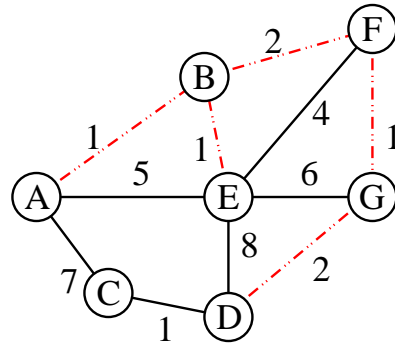
## STEP 3 IN DIJKSTRA's ALGORITHM



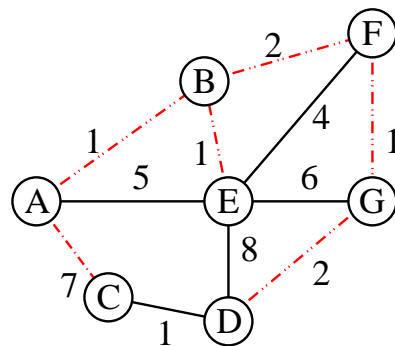
## STEP 4 IN DIJKSTRA's ALGORITHM



## STEP 5 IN DIJKSTRA's ALGORITHM



## STEP 6 IN DIJKSTRA's ALGORITHM

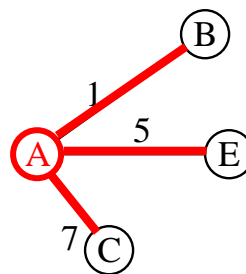


(b) using the Bellman-Ford algorithm (include the information about the predecessor node).

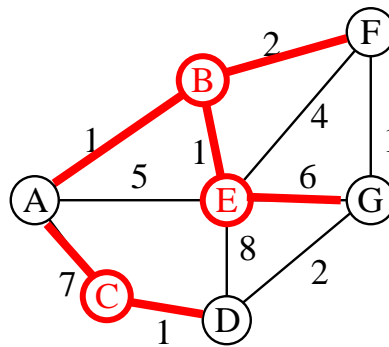
Answer:

The Bellman-Ford steps are:

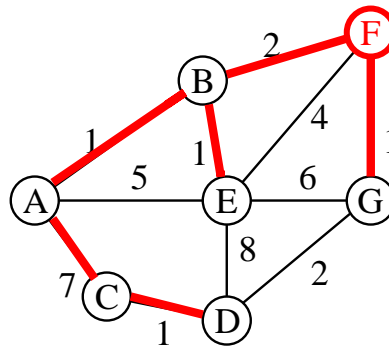
STEP 1:



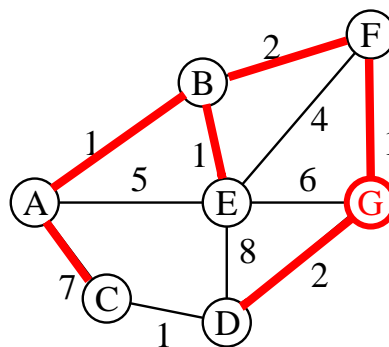
STEP 2:



STEP 3:



STEP 4:



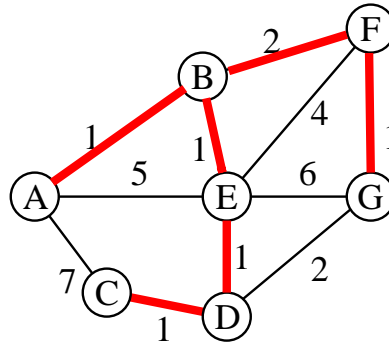
And the corresponding matrix:

| STEPS  | B          | C          | B          | E          | F          | G           |
|--------|------------|------------|------------|------------|------------|-------------|
| Start  | $\infty$   | $\infty$   | $\infty$   | $\infty$   | $\infty$   | $\infty$    |
| Step 1 | <u>1,A</u> | <u>7,A</u> | $\infty$   | <u>5,A</u> | $\infty$   | $\infty$    |
| Step 2 | 1,A        | 7,A        | <u>8,C</u> | <u>2,B</u> | <u>3,B</u> | <u>11,E</u> |
| Step 3 | 1,A        | 7,A        | 8,C        | <u>2,B</u> | <u>2,B</u> | <u>4,F</u>  |
| Step 4 | 1,A        | 7,A        | <u>6,G</u> | 2,B        | 2,B        | 4,F         |

- (c) Assume that after the Bellman-Ford algorithm completes (i.e., no further changes are made to the tree), the weight of edge  $(E, D)$  changes to 1. Continue the algorithm to find the new shortest path.

Answer:

The new shortest path: STEP 1:



And the new Bellman-Ford matrix:

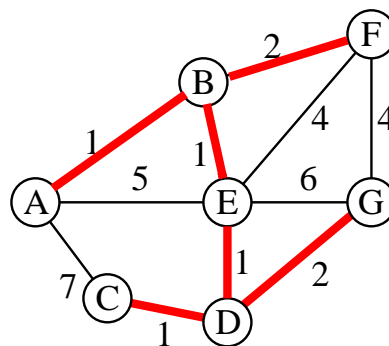
| STEPS    | B   | C          | B          | E   | F   | G   |
|----------|-----|------------|------------|-----|-----|-----|
| Previous | 1,A | 7,A        | 6,G        | 2,B | 2,B | 4,F |
| Step 1   | 1,A | <u>4,D</u> | <u>3,E</u> | 2,B | 2,B | 4,F |

- (d) Assume that after the Bellman-Ford algorithm completes for a second time, edge  $(F, G)$  changes its weight to 4. Continue the algorithm once more to find the new shortest path.

Answer:

The new shortest path:

STEP 1:



And the new Bellman-Ford matrix:

| STEPS    | B   | C   | B   | E   | F   | G          |
|----------|-----|-----|-----|-----|-----|------------|
| Previous | 1,A | 7,A | 6,G | 2,B | 2,B | 4,F        |
| Step 1   | 1,A | 4,D | 3,E | 2,B | 2,B | <u>5,D</u> |