CX2101 Algorithm Design and Analysis

Tutorial 2 (Graphs)

Week 7: Q4-Q6

This Tutorial

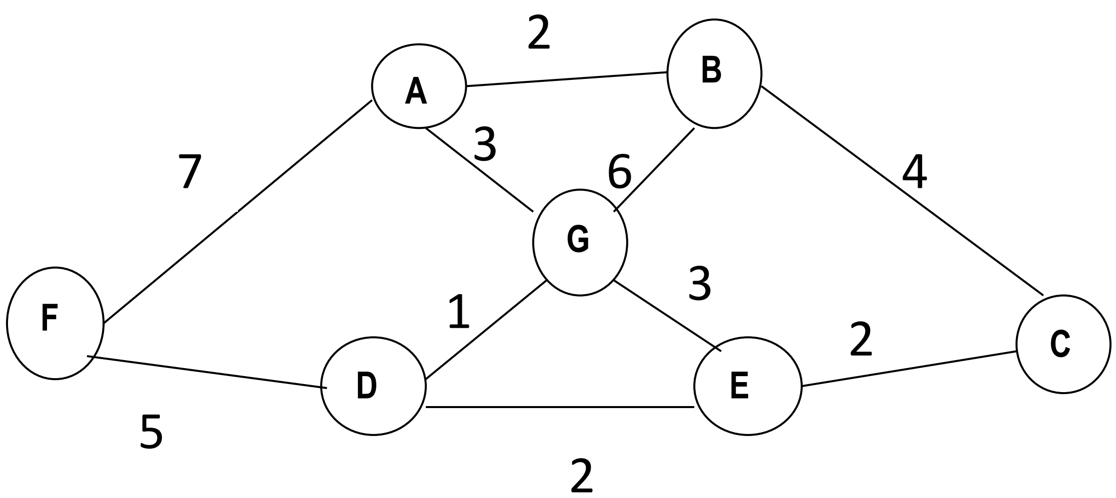
- Minimum spanning tree algorithm – Prim’s

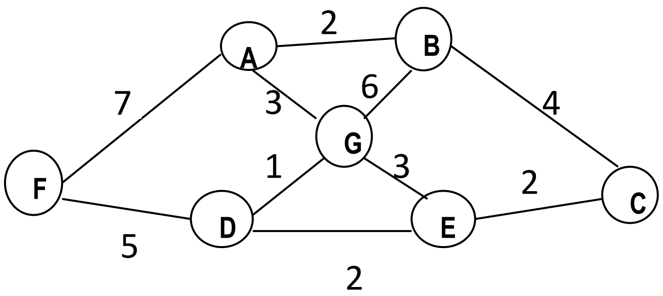
- Minimum spanning tree - concept

Question 4

Execute Prim's minimum spanning tree algorithm by hand on the graph below starting at vertex G.

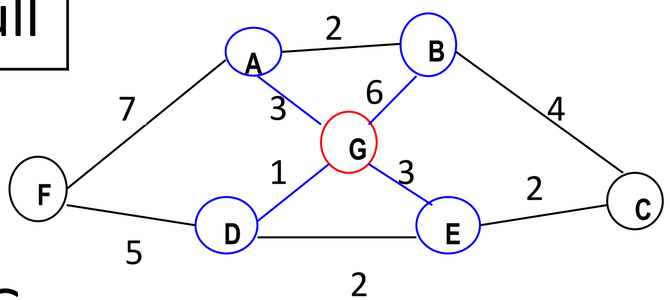
Show the contents of arrays S, d and pi after each iteration of the while loop when a vertex is added to the MST.





After initialization:

	A	B	C	D	E	F	G
S	0	0	0	0	0	0	1
d	**∞**	**∞**	**∞**	**∞**	**∞**	**∞**	**0**
pi	Null	Null	Null	Null	Null	Null	Null

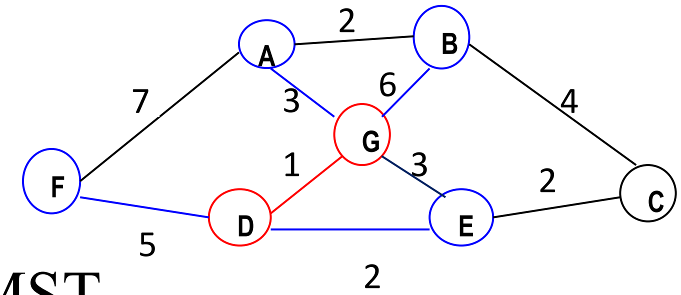


First iteration:

	A	B	C	D	E	F	G
S	0	0	0	0	0	0	1
d	**3**	**6**	**∞**	**1**	**3**	**∞**	**0**
pi	G	G	Null	G	G	Null	Null

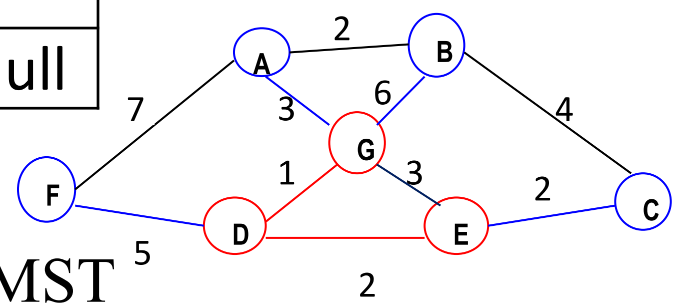
previous iteration →

	A	B	C	D	E	F	G
S	0	0	0	0	0	0	1
d	**3**	**6**	**∞**	**1**	**3**	**∞**	**0**
pi	G	G	Null	G	G	Null	Null



2nd iteration: Edge GD becomes part of the MST

	A	B	C	D	E	F	G
S	0	0	0	1	0	0	1
d	**3**	**6**	**∞**	**1**	**2**	**5**	**0**
pi	G	G	Null	G	D	D	Null

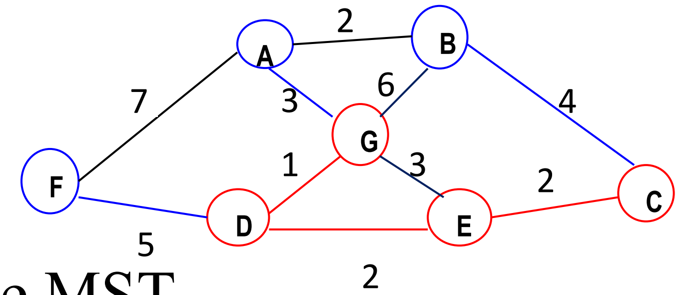


3rd iteration: Edge DE becomes part of the MST

	A	B	C	D	E	F	G
S	0	0	0	1	1	0	1
d	**3**	**6**	**2**	**1**	**2**	**5**	**0**
pi	G	G	E	G	D	D	Null

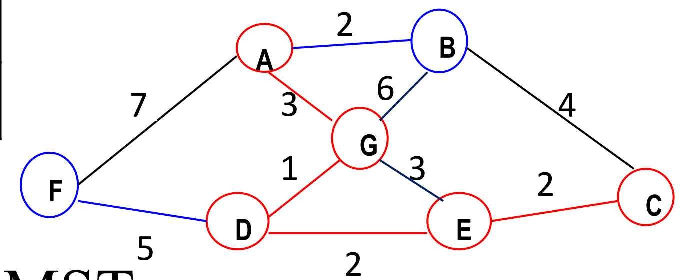
previous iteration →

	A	B	C	D	E	F	G
S	0	0	0	1	1	0	1
d	**3**	**6**	**2**	**1**	**2**	**5**	**0**
pi	G	G	E	G	D	D	Null



4th iteration: Edge EC becomes part of the MST

	A	B	C	D	E	F	G
S	0	0	1	1	1	0	1
d	**3**	**4**	**2**	**1**	**2**	**5**	**0**
pi	G	C	E	G	D	D	Null

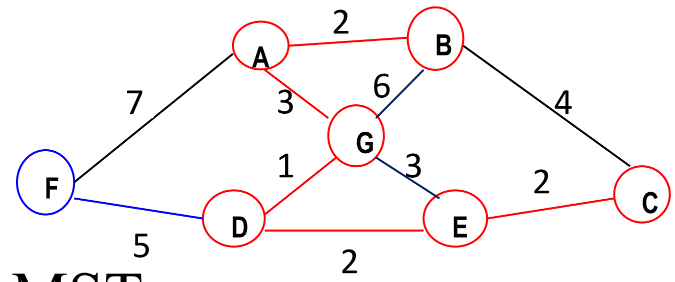


5th iteration: Edge GA becomes part of the MST

	A	B	C	D	E	F	G
S	1	0	1	1	1	0	1
d	**3**	**2**	**2**	**1**	**2**	**5**	**0**
pi	G	A	E	G	D	D	Null

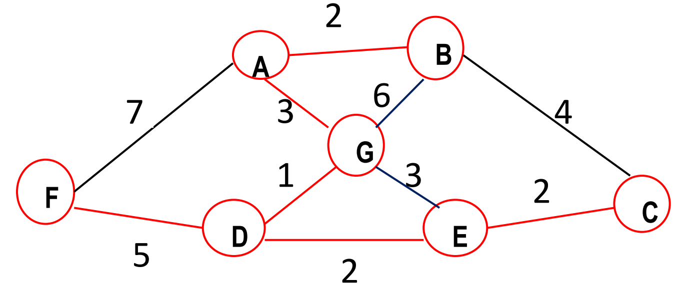
previous iteration

	A	B	C	D	E	F	G
S	1	0	1	1	1	0	1
d	**3**	**2**	**2**	**1**	**2**	**5**	**0**
pi	G	A	E	G	D	D	Null



6th iteration: Edge AB becomes part of the MST

	A	B	C	D	E	F	G
S	1	1	1	1	1	0	1
d	**3**	**2**	**2**	**1**	**2**	**5**	**0**
pi	G	A	E	G	D	D	Null



7th iteration: Edge DF becomes part of the MST

	A	B	C	D	E	F	G
S	1	1	1	1	1	1	1
d	**3**	**2**	**2**	**1**	**2**	**5**	**0**
pi	G	A	E	G	D	D	Null

Question 5

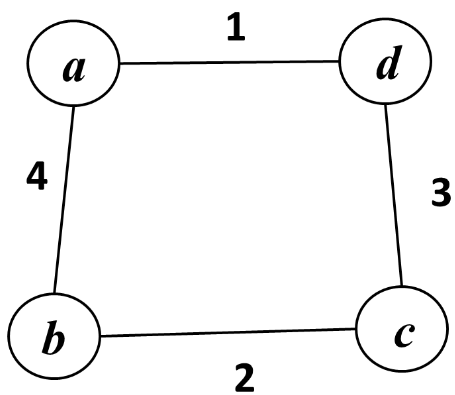
In a weighted undirected graph, is the path between two vertices in a minimum spanning tree always the shortest path (i.e. a path with the minimum weight) between the two vertices in the graph?

If your answer is yes, give a proof;

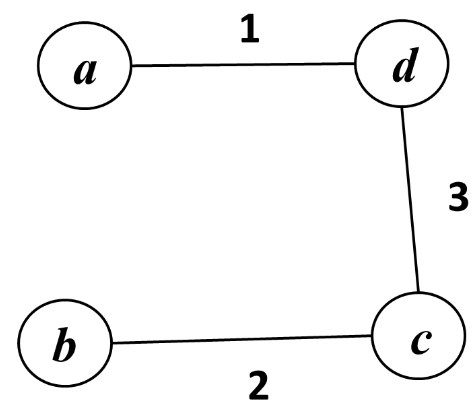
otherwise, give a counterexample.

Answer: No, it is not always the case.

A counterexample:



A weighted graph



Minimum Spanning Tree (MST)

- The path from vertex *a* to vertex *b* in the MST is (*a, d, c, b*) with weight 1 + 3 + 2 = 6.
- But the shortest path from *a* to *b* in the graph is (*a, b*) with weight 4, shorter than the path in the MST.

Question 6

Draw a connected graph with five nodes, six edges of respective weights 5, 6, 7, 8, 9, 10, and a minimum spanning tree of weight 28.

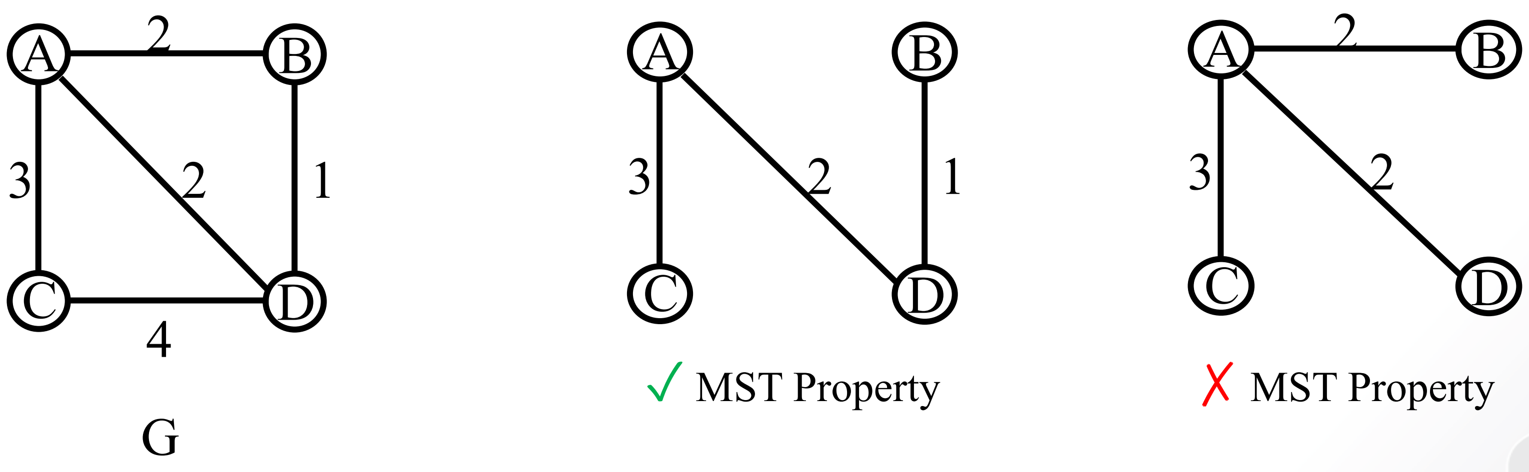
Is it possible to have an MST of weight 29?

- If yes, draw the graph;
- otherwise, provide your justification.

MST Property

Minimum Spanning Tree Property

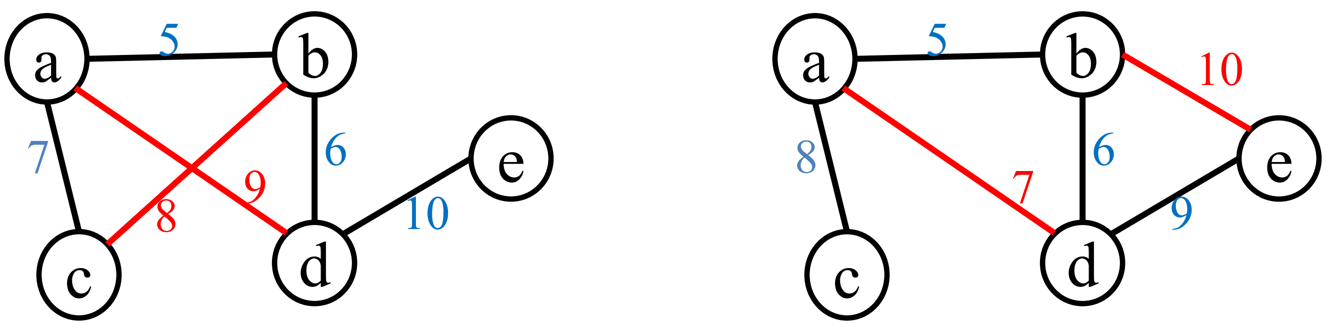
Let T be a spanning tree of G, where $G = (V, E, W)$ is a connected, weighted graph. Suppose that for every edge (u, v) of G that is not in T, if (u, v) is added to T it creates a cycle such that (u, v) is a maximum-weight edge on that cycle. Then T has the **Minimum Spanning Tree Property** (or **MST Property**, in short).



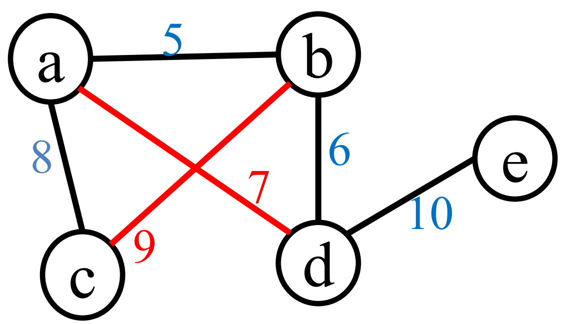
Theorem 1

Theorem 1: In a connected weighted graph G = (V, E, W), a tree *T* is a minimum spanning tree if and only if *T* has the MST property.

- Edge weights: 5, 6, 7, 8, 9, 10
- MST of weight 28: 5, 6, 7, 10 or 5, 6, 8, 9



- Edge weights: 5, 6, 7, 8, 9, 10
- MST of weight 29: 5, 6, 8, 10



What we have exercised

- Minimum spanning tree algorithm
  - Prim’s algorithm
    - Running of the algorithm
    - Data structure contents in each iteration
    - Path in MST may not be a shortest path in G
- Minimum spanning tree – concept
  - Understand which edges may or may not be in an MST