ASSIGNMENT 4

MERT ZİYA
27800

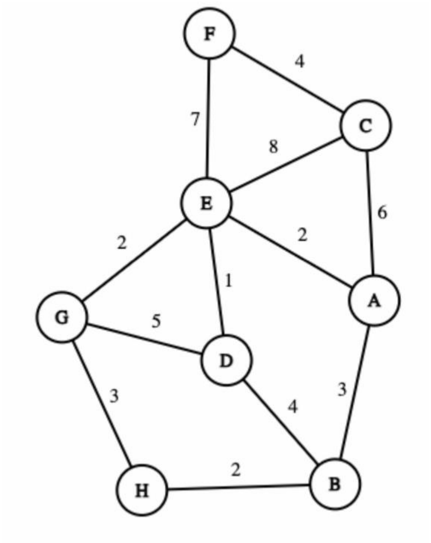


Figure 1: An undirected weighted graph.

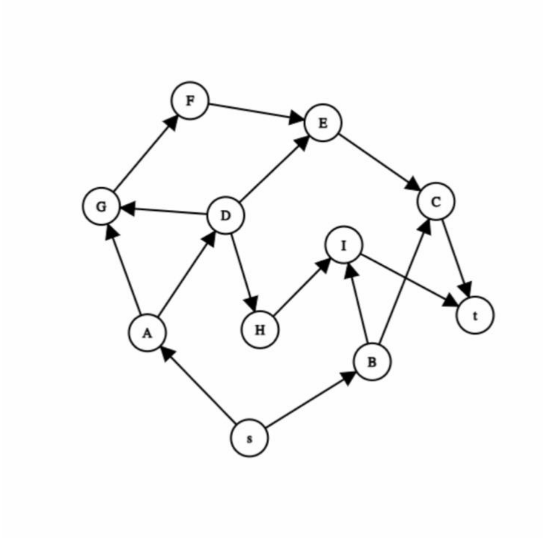


Figure 2: A directed acyclic graph.

QUESTION 1:

"Trace the Dijkstra's weighted shortest path algorithm on the graph given in Figure 1. Use vertex E as your start vertex. ":

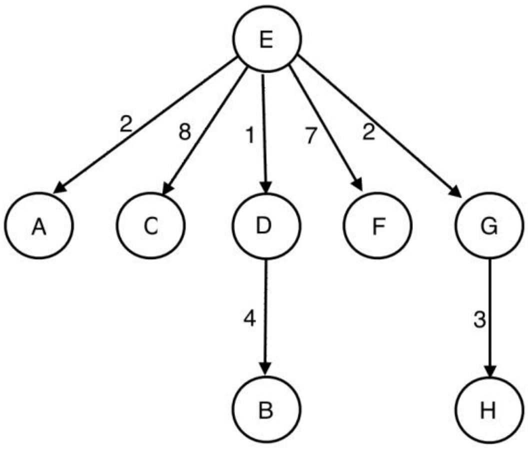
V	A	B	C	D	E	F	G	H
E	(2,E,U)	(∞,?,U)	(8,E,U)	(1,E,U)	(0,E,S)	(7,E,U)	(2,E,U)	(∞,?,U)
D	(2,E,U)	(5,D,U)	(8,E,U)	(1,E,S)	(0,E,S)	(7,E,U)	(2,E,U)	(∞,?,U)
A	(2,E,S)	(5,D,U)	(8,E,U)	(1,E,S)	(0,E,S)	(7,E,U)	(2,E,U)	(∞,?,U)
G	(2,E,S)	(5,D,U)	(8,E,U)	(1,E,S)	(0,E,S)	(7,E,U)	(2,E,S)	(5,G,U)
B	(2,E,S)	(5,D,S)	(8,E,U)	(1,E,S)	(0,E,S)	(7,E,U)	(2,E,S)	(5,G,U)
H	(2,E,S)	(5,D,S)	(8,E,U)	(1,E,S)	(0,E,S)	(7,E,U)	(2,E,S)	(5,G,S)
F	(2,E,S)	(5,D,S)	(8,E,U)	(1,E,S)	(0,E,S)	(7,E,S)	(2,E,S)	(5,G,S)
C	(2,E,S)	(5,D,S)	(8,E,S)	(1,E,S)	(0,E,S)	(7,E,S)	(2,E,S)	(5,G,S)

- (x,y,z) Elements inside the table is in the following format:
  - x = path length from the vertex E
  - y = the vertex coming before that current vertex
  - z = Selected (S) or Unselected (U)

notes:

1) If there are 2 same values after implementing one step, any vertex can be selected as an updated vertex it doesn't matter
2) İf an equal path length in a step is encountered, don't update it (because there is no need for that).

Tree representation of shortest path length from E after implementing Dijkstra's algorithm;



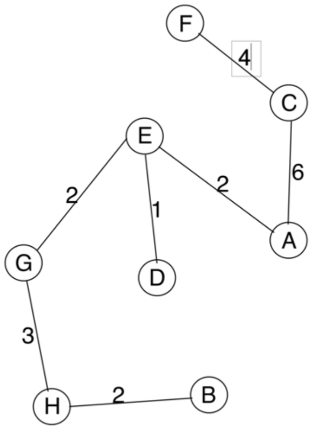
QUESTION 2:

"Trace the Prim's minimum spanning tree algorithm on the graph in Figure 1. Use vertex E as your start vertex.":

Known Vertices	Edge, that has the smallest weight AND connected to a known vertex AND unknown
{E}	(E,D,1)
{E,D}	(E,G,2)
{E,D,G}	(E,A,2)
{E,D,G,A}	(G,H,3)
{E,D,G,A,H}	(H,B,2)
{E,D,G,A,H,B}	(A,C,6)
{E,D,G,A,H,B,C}	(C,F,4)
{E,D,G,A,H,B,C,F}	-

<u>Note:</u> if there are equal weighted edges in a step any edge can be selected, it would not matter.

Minimum Spanning Tree after implementing the Prim's Algorithm:



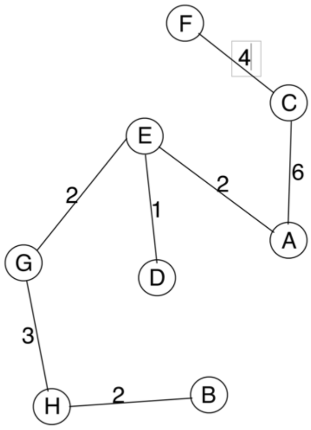
QUESTION 3:

Trace the Kruskal's minimum spanning tree algorithm on the graph in Figure 1:

Unselected Edges	Edge with smallest weight among unselected edges which does not forms a cycle
{ (E,D,1), (E,G,2), (E,A,2), (H,B,2), (G,H,3), (B,A,3), (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8) }	(E,D,1)
{ (E,G,2), (E,A,2), (H,B,2), (G,H,3), (B,A,3), (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8)}	(E,G,2)
{ (E,A,2), (H,B,2), (G,H,3), (B,A,3), (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8)}	(E,A,2)
{ (H,B,2), (G,H,3), (B,A,3), (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8)}	(H,B,2)
{ (G,H,3), (B,A,3), (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8)}	(G,H,3)
{ (B,A,3), (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8) }	(B,A,3) -> discarded
{ (D,B,4), (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8) }	(D,B,4) -> discarded
{ (F,C,4), (G,D,1), (A,C,6), (E,F,7), (E,C,8) }	(F,C,4)
{ (G,D,1), (A,C,6), (E,F,7), (E,C,8) }	(G,D,5) -> discarded
{ (A,C,6), (E,F,7), (E,C,8) }	(A,C,6)
{ (E,F,7), (E,C,8) }	(E,F,7) -> discarded
{ (E,C,8) }	(E,C,8) -> discarded
-	-

<u>Note:</u> if 2 or more edge has the same weight select a random edge it would not matter.
<u>Note:</u> If an edge causes a cycle in the minimum spanning tree mark it as select it but dont include it in the minimum spanning tree (discard it).

Minimum Spanning Tree after the implementation of Kruskal's Algorithm:



QUESTION 4:

Trace the breadth-first search traversal algorithm on the graph in Figure 1 starting from vertex E:

Considering the Vertices	Information
E	Path=E , Distance = 0, Known = True
G	Path=E , Distance = 2, Known = True
D	Path=E , Distance = 1, Known = True
A	Path=E , Distance = 2, Known = True
C	Path=E , Distance = 8, Known = True
F	Path=E , Distance = 7, Known = True
H	Path=G , Distance = (2+2), Known = True
B	Path=D , Distance = (1+4), Known = True

Note: this table is not unique, there can be different tables as a result when BFS algorithm is implemented to the graph.

QUESTION 5:

Find a topological ordering of the graph in Figure 2:

Number of Vertices in Graph	**Vertices with indegree = 0**	**Selected vertices among the vertices with indegree = 0**
11	{s}	s
10	{A,B}	A
9	{B,D}	B
8	{D}	D
7	{G,H}	G
6	{H,F}	H
5	{F,I}	F
4	{I,E}	I
3	{E}	E
2	{C}	C
1	{t}	t
0	{}	-

Note: Topological sort doesn't give a unique order. As a result, graph can be ordered in many ways.

*→ Topological order after **this** way:*
(s,A,B,D,G,H,F,I,E,C,t)