1)

1. Known nodes:
   Unknown nodes: S (?, ∞), A (?, ∞), B (?, ∞), C (?, ∞), D (?, ∞), E (?, ∞), F (?, ∞), G (?, ∞)

2. Known nodes: **S(S, 0)**
   Unknown nodes: A (S, 3), B (S, 2), C (?, ∞), D (?, ∞), E (?, ∞), F (?, ∞), G (?, ∞)

3. Known nodes: S(S, 0), **B(S, 2)**
   Unknown nodes: A(S, 3), C (?, ∞), D (B, 3), E (?, ∞), F (?, ∞), G (?, ∞)

4. Known nodes: S(S, 0), B(S, 2), **A(S, 3)**
   Unknown nodes: C (?, ∞), D (B, 3), E (A, 5), F (?, ∞), G (?, ∞)

5. Known nodes: S(S, 0), B(S, 2), A(S, 3), **D(B, 3)**
   Unknown nodes: C (?, ∞), E (D, 4), F (D, 5), G (D, 5)

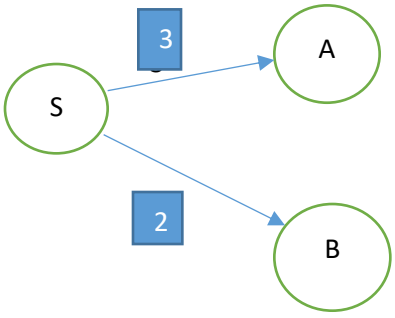
6. Known nodes: S(S, 0), B(S, 2), A(S, 3), D(B, 3), **E(D, 4)**
   Unknown nodes: C (?, ∞), F (D, 5), G (D, 5)

7. Known nodes: S(S, 0), B(S, 2), A(S, 3), D(B, 3), E(D, 4), **G(D, 5)**
   Unknown nodes: C (?, ∞), F (D, 5)

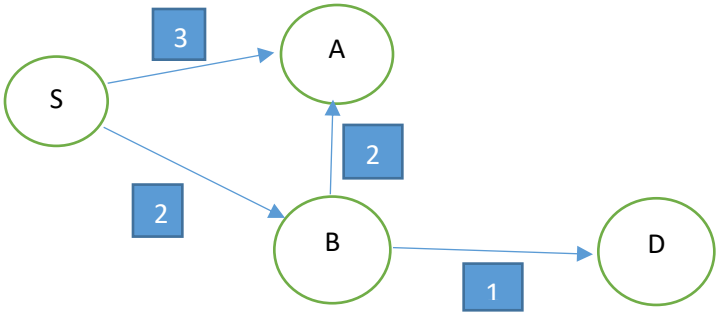
8. Known nodes: S(S, 0), B(S, 2), A(S, 3), D(B, 3), E(D, 4), G(D, 5), **F(D, 5)**
   Unknown nodes: C (F, 6)

9. Known nodes: S(S, 0), B(S, 2), A(S, 3), D(B, 3), E(D, 4), G(D, 5), F(D, 5), **C(F, 6)**
   Unknown nodes:

2) We begin with the starting vertex S and then grow the tree in successive order. At each stage, a new edge (u, v) will be added to the tree if edge (u, v) has the smallest cost among all edges such that u is in tree and v is not. For example, we compare the adjacent edges of S which are: (S, B) and (S, A) with weights 2 and 3. Since (S, B) has a smaller weight so it is picked and we continue the same process in successive order. Edges that create cycles are avoided.

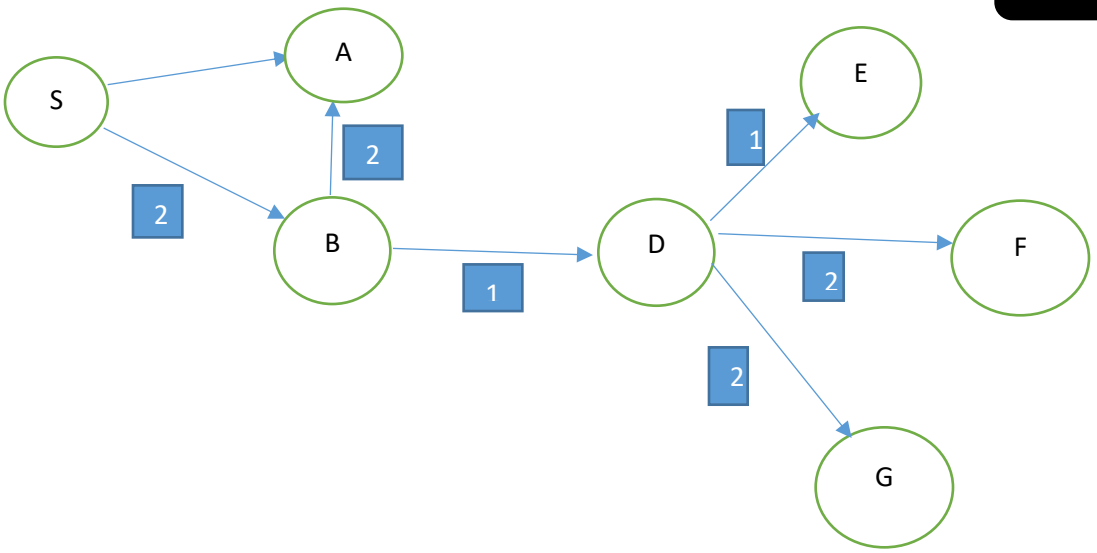


V = (S,)

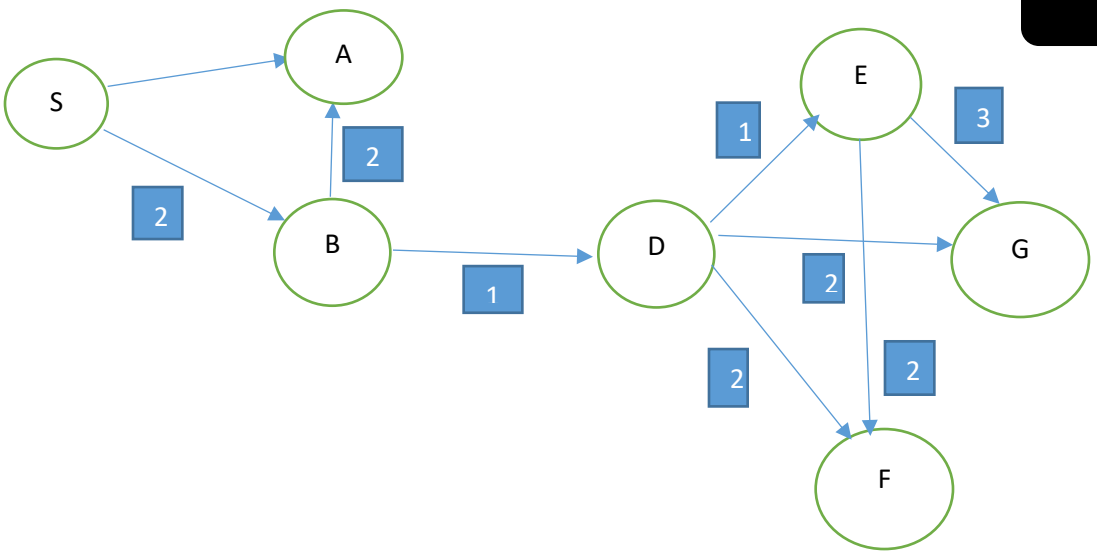


V = (S, B)

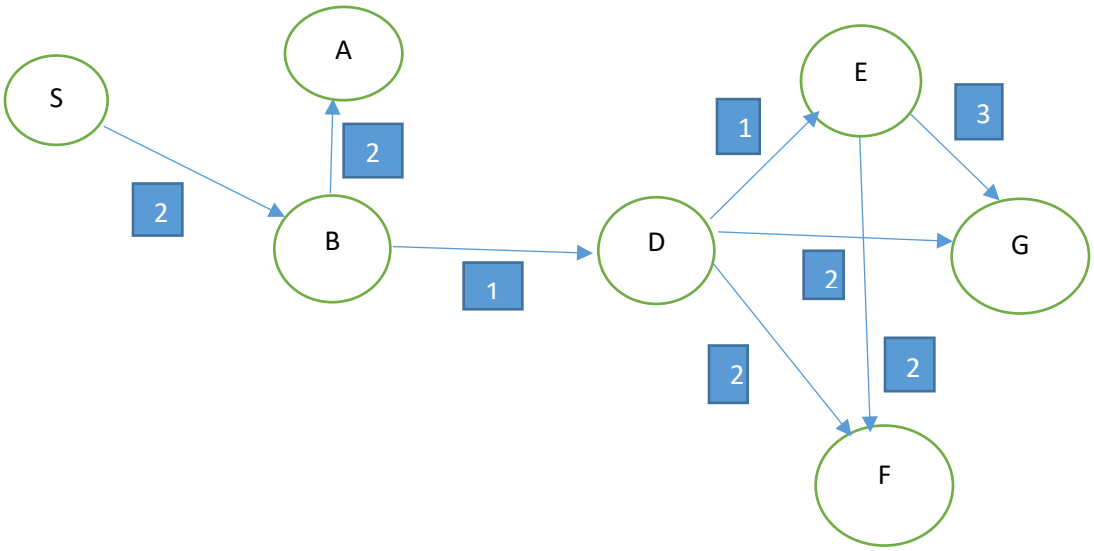
V = (S, B, D)



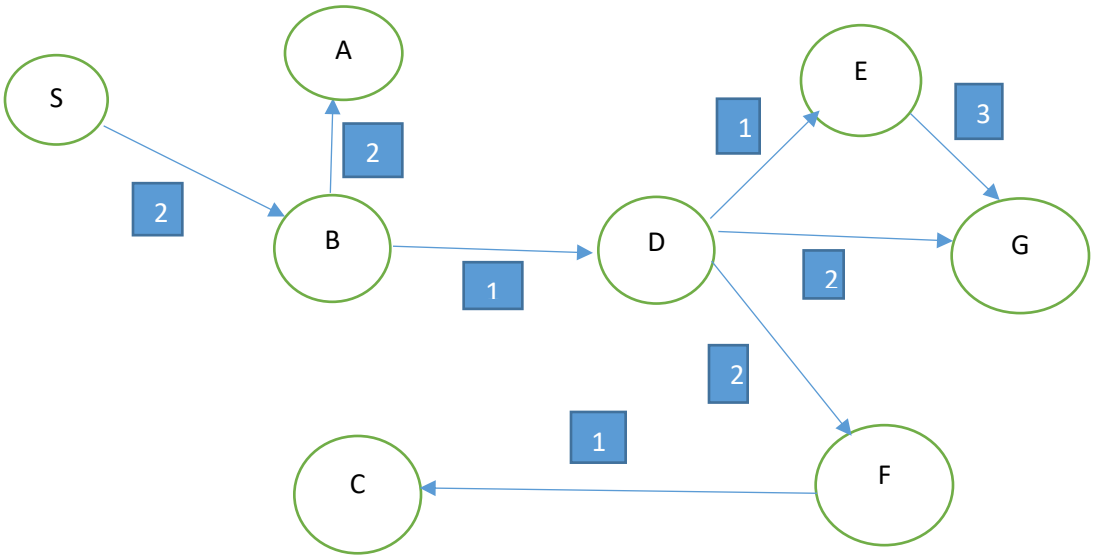
V = (S, B, D, E)



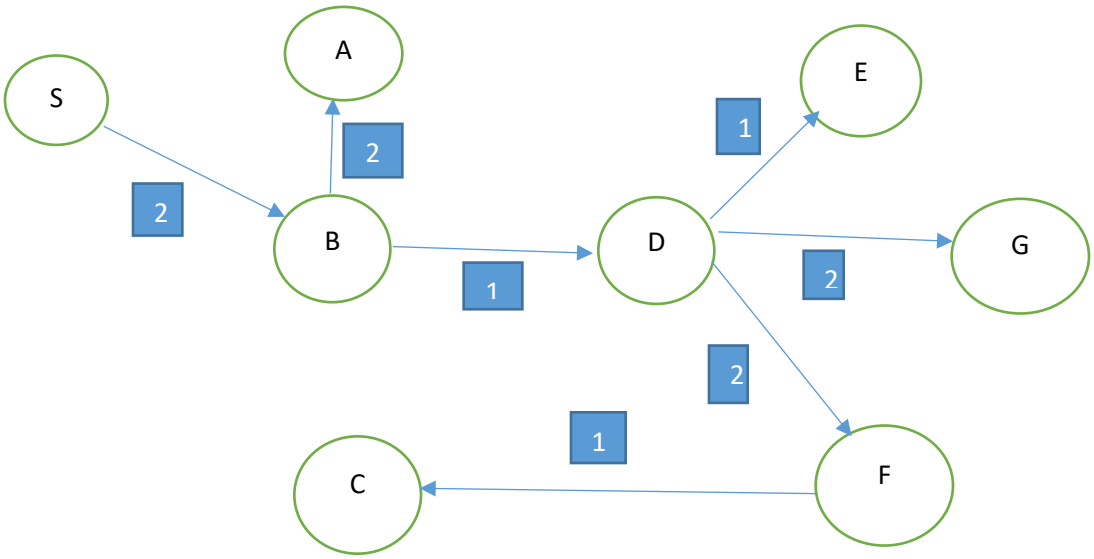
V = (S, B, D, E, A)



V = (S, B, D, E, A, F, C)

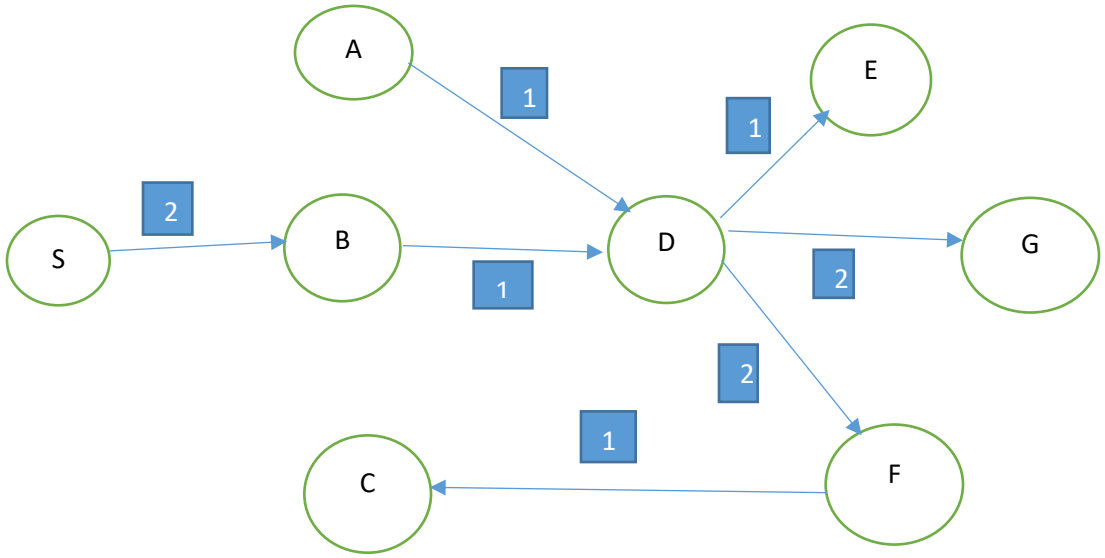


V = (S, B, D, E, A, F, C, G)



3)

- Pick edge (A,D) Tree = A, D
- Pick edge (B,D) Tree = A, D, B
- Pick edge (D,E) Tree = A, D, B, E
- Pick edge (F,C) Tree = A, D, B, E Tree 2 = F, C
- Pick edge (S,B) Tree = A, D, B, E, S Tree 2 = F, C
- Pick edge (A,E) forms cycle, not included
- Pick edge (A,B) forms cycle, not included
- Pick edge (D,G) Tree = A, D, B, E, S , G Tree 2 = F, C
- Pick edge (D,F) Tree = A, D, B, E, S , G, F, C

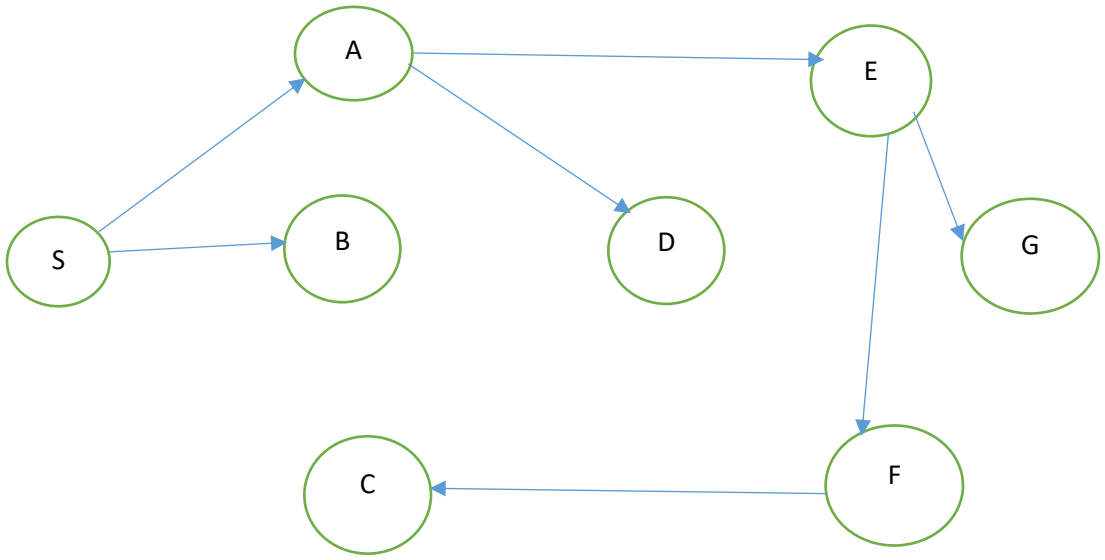


4) Starting from node S, Q in the queue and V is set of visited nodes.

- Q = {S}
  V = {}
- Q = {A, B}
  V = {S}
- Q = {B, E, D}
  V = {S, A}
- Q = {E, D}
  V = {S, A, B}
- Q = {D, G, F}
  V = {S, A, B, E}
- Q = {G, F}
  V = {S, A, B, E, D}
- Q = { F}
  V = {S, A, B, E, D, G}

- Q = {C}
  V = {S, A, B, E, D, G, F}

- Q = {}
  V = {S, A, B, E, D, G, F, C}



5)

a) The order will be:

S → B → D → E → G →F →C →A

b)

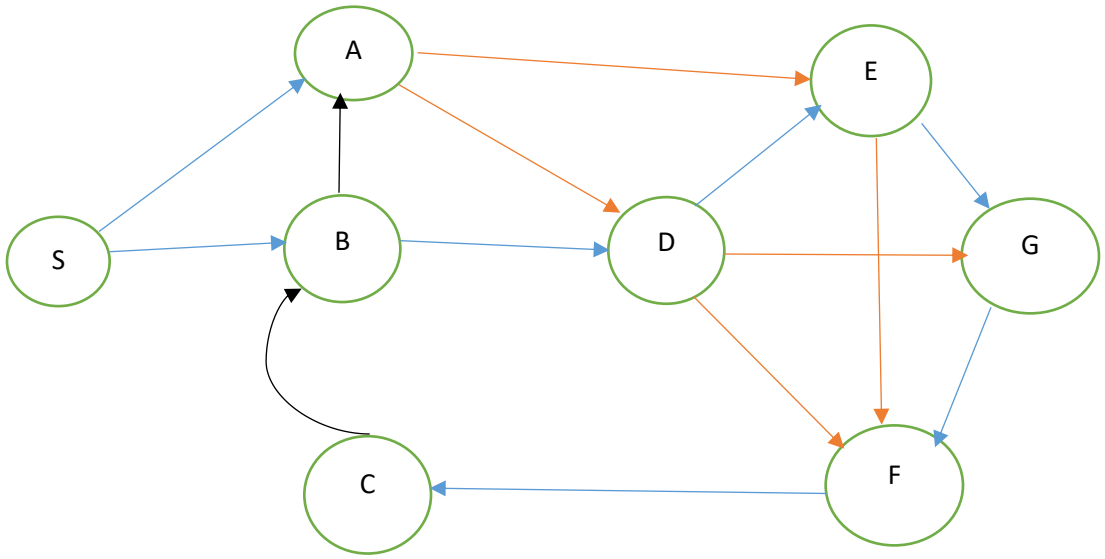
C	1
F	2
G	3
E	4
D	5
B	6
A	7
S	8

c)

S	1
B	2
D	3
E	4
G	5
F	6
C	7
A	8

d)

- Tree arcs: **SB, SA, BD, DE, EG, GF, FC** (blue line)
- Forward arcs: **AE, DF, EF, DG, AD** (orange line)
- Backward arcs: **CB** (curly block line)
- Cross arcs: **BA** (black line)



6) Possible topological ordering: **D, B, A, E, F, C, G**