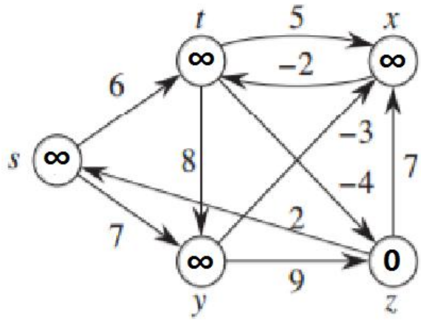
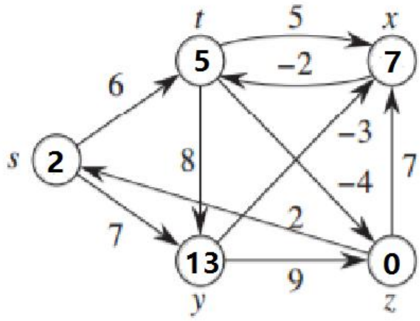
1.

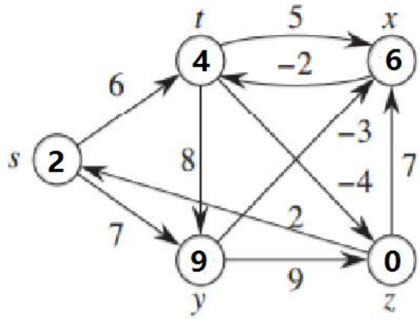
Initial:



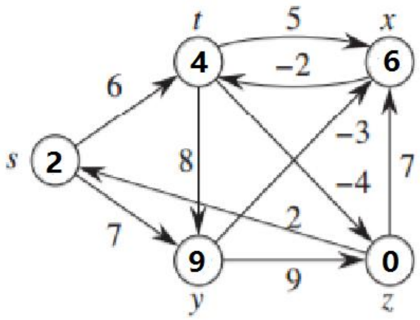

First Pass:



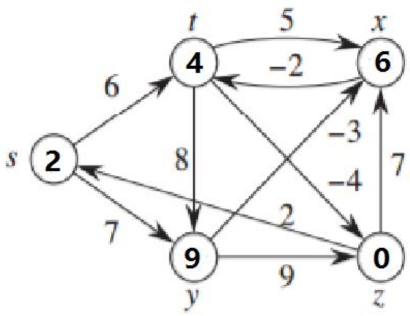

Second Pass:



Third Pass:




Fourth Pass:



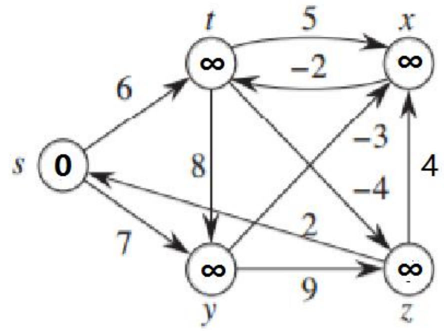

| d | s | t | x | y | z |
|---|---|---|---|---|---|
| | ∞ | ∞ | ∞ | ∞ | 0 |
| 1 | 2 | 5 | 7 | 13 | 0 |
| 2 | 2 | 4 | 6 | 9 | 0 |
| 3 | 2 | 4 | 6 | 9 | 0 |
| 4 | 2 | 4 | 6 | 9 | 0 |

| π | s | t | x | y | z |
|---|---|---|---|---|---|
| | NIL | NIL | NIL | NIL | NIL |
| 1 | z | x | z | t | NIL |
| 2 | z | x | y | s | NIL |
| 3 | z | x | y | s | NIL |
| 4 | z | x | y | s | NIL |

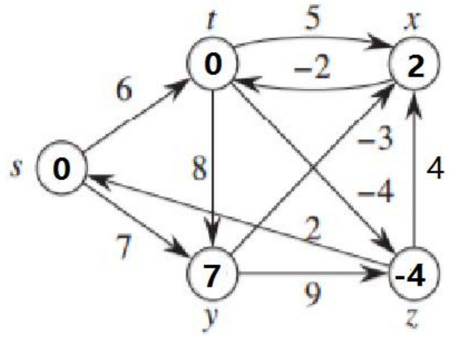
Return value is True.

Change wright of (z, x) to 4.

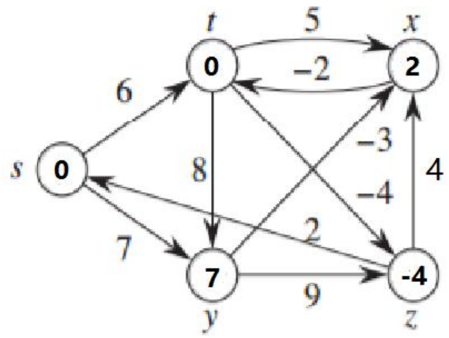
Initial:



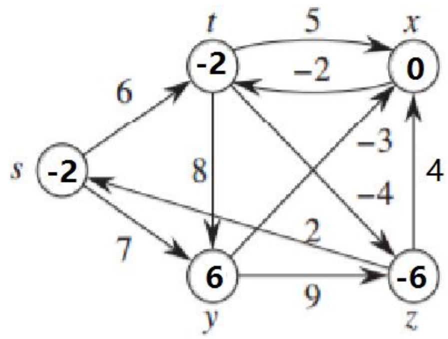

First Pass:



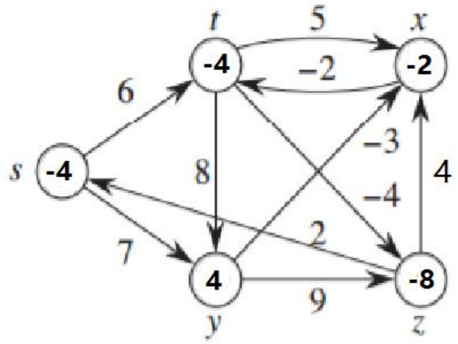

Second Pass:



Third Pass:




Fourth Pass:

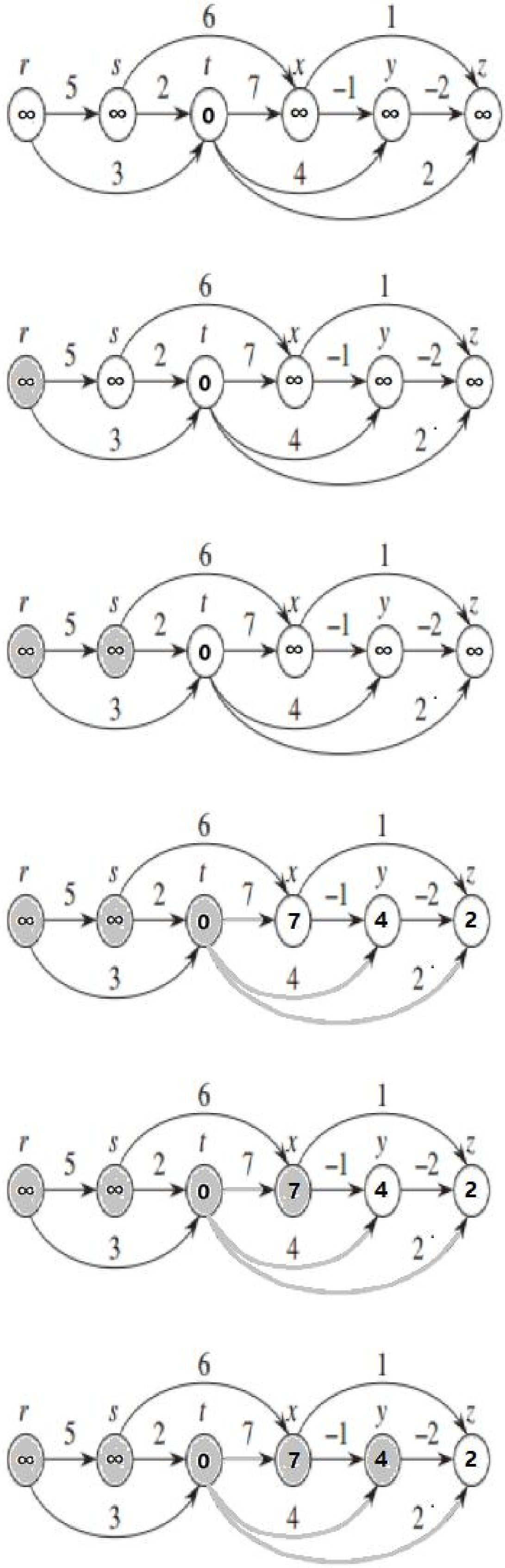



| d | s | t | x | y | z |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 2 | 4 | 7 | -2 |
| 2 | 0 | 0 | 2 | 7 | -4 |
| 3 | -2 | -2 | 0 | 6 | -6 |
| 4 | -4 | -4 | -2 | 4 | -8 |

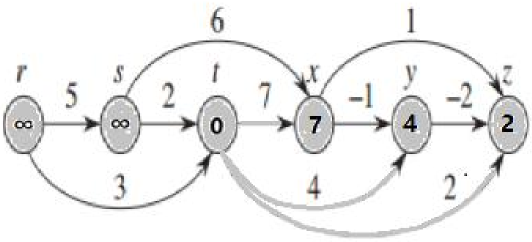
| π | s | t | x | y | z |
|---|---|---|---|---|---|
| | NIL | NIL | NIL | NIL | NIL |
| 1 | NIL | x | y | s | t |
| 2 | NIL | x | x | s | t |
| 3 | z | x | x | s | t |
| 4 | z | x | x | s | t |

Return value is False.

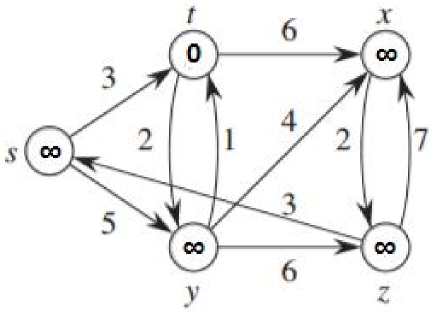
**2.**

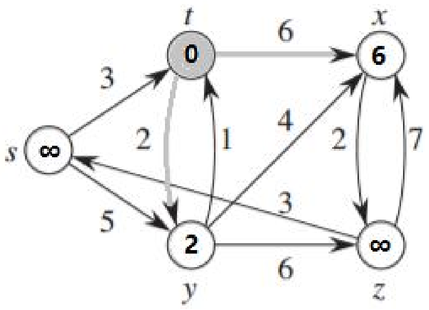


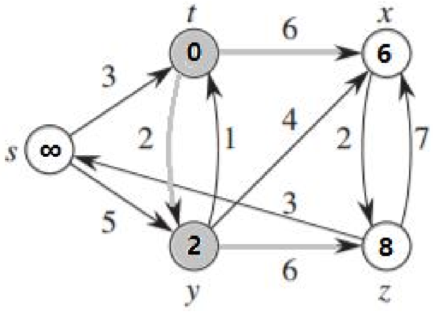
6 1
r s t x y z
5 2 7 −1 −2
∞ ∞ 0 ∞ ∞ ∞
3 4 2
6 1
r s t x y z
5 2 7 −1 −2
∞ ∞ 0 ∞ ∞ ∞
3 4 2
6 1
r s t x y z
5 2 7 −1 −2
∞ ∞ 0 ∞ ∞ ∞
3 4 2
6 1
r s t x y z
5 2 7 −1 −2
∞ ∞ 0 7 4 2
3 4 2
6 1
r s t x y z
5 2 7 −1 −2
∞ ∞ 0 7 4 2
3 4 2
6 1
r s t x y z
5 2 7 −1 −2
∞ ∞ 0 7 4 2
3 4 2

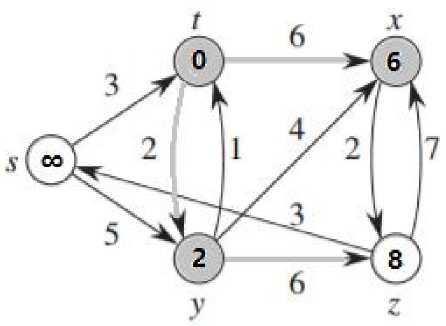

r s t x y z
∞ ∞ 0 7 4 2
5 2 7 –1 –2
6 1
3 4 2

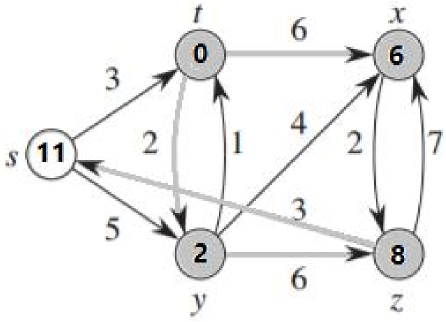
3.

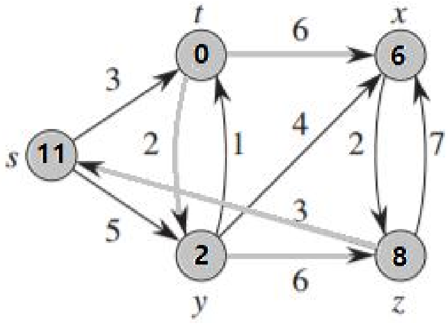

t x s y z
0 ∞ ∞ ∞ ∞
6 3 2 1 4 2 7 3 5 6


t x s y z
0 6 ∞ 2 ∞
6 3 2 1 4 2 7 3 5 6


t x s y z
0 6 ∞ 2 8
6 3 2 1 4 2 7 3 5 6


t x s y z
0 6 ∞ 2 8
6 3 2 1 4 2 7 3 5 6




| d | s | t | x | y | z |
|---|---|---|---|---|---|
| | ∞ | 0 | ∞ | ∞ | ∞ |
| 1 | ∞ | 0 | 6 | 2 | ∞ |
| 2 | ∞ | 0 | 6 | 2 | 8 |
| 3 | ∞ | 0 | 6 | 2 | 8 |
| 4 | 11 | 0 | 6 | 2 | 8 |
| 5 | 11 | 0 | 6 | 2 | 8 |

| π | s | t | x | y | z |
|---|---|---|---|---|---|
| | NIL | NIL | NIL | NIL | NIL |
| 1 | NIL | x | y | s | t |
| 2 | NIL | x | x | s | t |
| 3 | NIL | x | x | s | t |
| 4 | z | x | x | s | t |
| 5 | z | x | x | s | t |

4.

```
RELIABILITY(G, r, x, y)
    INITIALIZE-SINGLE-SOURCE(G, x)  Initialize to be negative infinite
    S = ∅
    Q = G.V
    while Q ≠ ∅ do
        u = EXTRACT- MAX(Q)
        S = S ∪ {u}
        for each vertex v ∈ G.Adj[u] do
            if v.d > u.d · r(u, v) then
                v.d = u.d · r(u, v)
                v.π = u
            end if
        end for
    end while
    while y ≠ x do
        Print y
        y = y.π
    end while
    Print x
```

5.

```
MODIFIED-DIJKSTRA(G,w,s)
for each v ∈ G.V do v.d = VW + 1 v.π = NIL
end for
s.d = 0
Initialize an array A of length VW + 2
A[0].insert(s)
Set A[VW + 1] equal to a linked list containing every vertex except s
k = 0
for i = 1 to |V| do
    while A[k] = NIL do
        k = k + 1
    end while
    u = A[k].head
    A[k].delete(u)
    for each vertex v ∈ G.Adj[u] do
        if v.d > u.d + w(u, v) then
            A[v.d].delete(v.list)
            v.d = u.d + w(u, v)
            v.π = u
            A[v.d].insert(v)
            v.list = A[v.d].head
        end if
    end for
end for
```

6.

| $k$ | $D^k$ |
|---|---|
| 0 | $\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & \infty & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & 3 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$ |
| 1 | $\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ \infty & 2 & 0 & \infty & \infty & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ \infty & 7 & \infty & \infty & 0 & \infty \\ \infty & 5 & 10 & \infty & \infty & 0 \end{pmatrix}$ |

| | |
|---|---|
| 2 | $\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$ |
| 3 | $\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ 1 & 0 & \infty & 2 & 0 & \infty \\ 3 & 2 & 0 & 4 & 2 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 8 & 7 & \infty & 9 & 0 & \infty \\ 6 & 5 & 10 & 7 & 5 & 0 \end{pmatrix}$ |
| 4 | $\begin{pmatrix} 0 & \infty & \infty & \infty & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & \infty & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$ |
| 5 | $\begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ 0 & 2 & 0 & 4 & -1 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$ |
| 6 | $\begin{pmatrix} 0 & 6 & \infty & 8 & -1 & \infty \\ -2 & 0 & \infty & 2 & -3 & \infty \\ -5 & -3 & 0 & -1 & -6 & -8 \\ -4 & 2 & \infty & 0 & -5 & \infty \\ 5 & 7 & \infty & 9 & 0 & \infty \\ 3 & 5 & 10 & 7 & 2 & 0 \end{pmatrix}$ |

## 7.

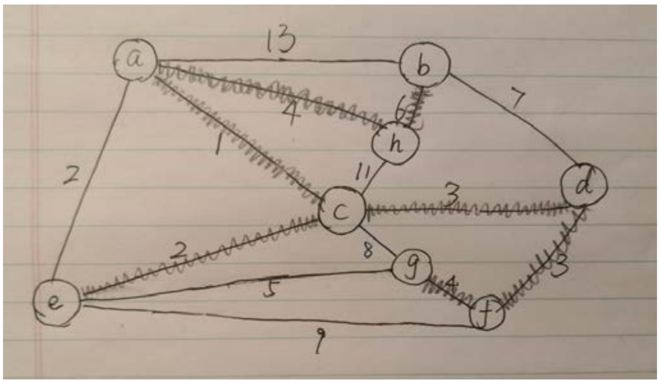
FLOYD-WARSHALL(W)
n = W.rows
$D^{(0)}$ = W
**for** k = 1 to n
    **let** $D^{(k)} = (d_{ij}^{(k)})$ , $\Phi^{(k)} = (\Phi_{ij}^{(k)})$be a new n * n matrix
    **for** i = 1 **to** n
        **for** j = 1 **to** n

        **if** $d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$
            $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$, $\Phi_{ij}^{(k)} = k$
        **else**
            $d_{ij}^{(k)} = d_{ij}^{(k-1)}$ , $\Phi_{ij}^{(k)} = \Phi_{ij}^{(k-1)}$
**return** $D^{(n)}$

```
PRINT-ALL-PAIRS-SHORTEST-PATH(Φ, i, j)
    if i == j
        print i
    elseif Φ_ij == NIL
        print "no path between i and j"
    else
        PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, Φ_ij)
        print Φ_ij
        PRINT-ALL-PAIRS-SHORTEST-PATH(Π, Φ_ij, j)
```

8.

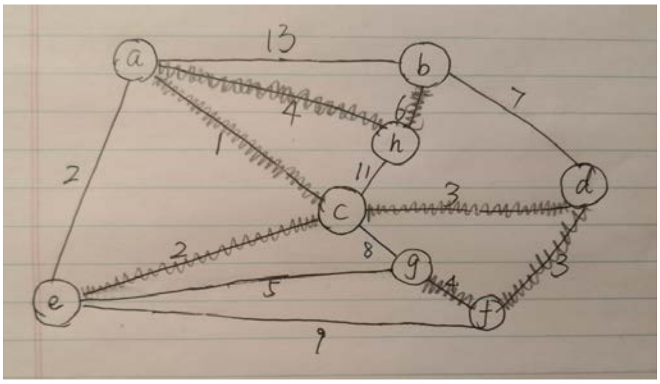
We can get one of the MST shown as follows:

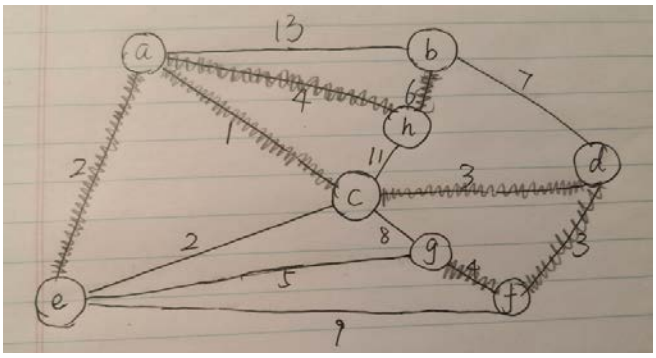



So, the cost is 1 + 2 + 3 + 3 + 4 + 4 + 6 = 23.

(b)

We can get 2 different MST:




(c)

For the second MST above, the order of the edges is (a, c), (a, e), (d, f), (c, d), (a, h), (f, g), (b, h).

For (a, c), We have S = {a}, V – S = {b, c, d, e, f, g, h}, so we add a safe edge (a, c) that w(a, c) = 1. And we will get A = {(a, c)}.

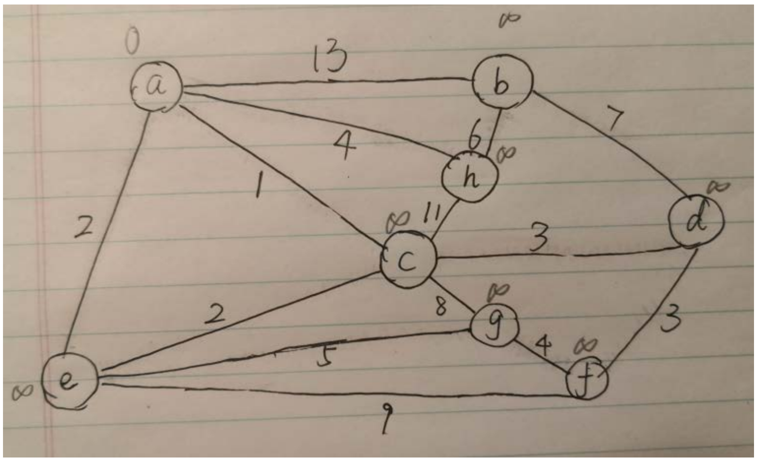
For (a, e), We have S = {a, c}, V – S = {b, d, e, f, g, h}, so we add a safe edge (a, e) that w(a, e) =

2. And we will get A = {(a, c), (a, e)}.

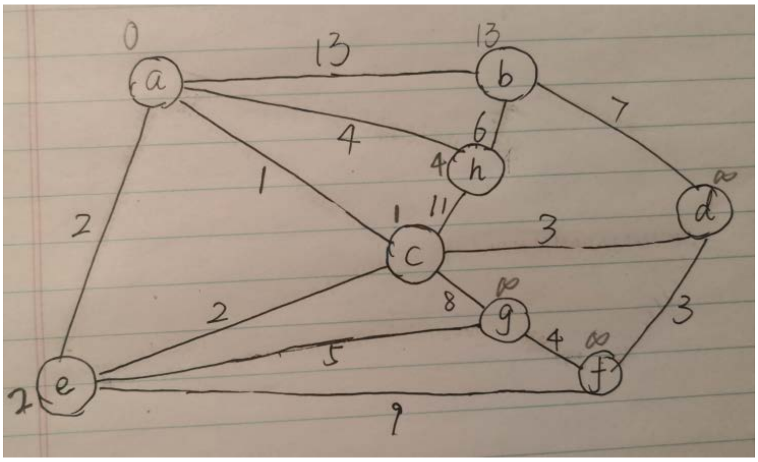
For (d, f), We have S = {d}, V – S = {a, b, c, e, f, g, h}, so we add a safe edge (d, df that w(d, f) = 3. And we will get A = {(a, c), (a, e), (d, f)}.

9.

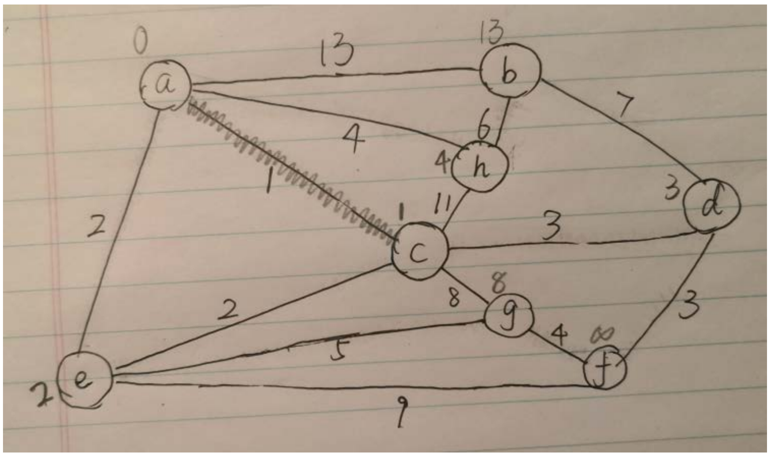
At the beginning: A = {}, Q = {a, b, c, d, e, f, g, h} and a.key = 0, other keys are ∞.



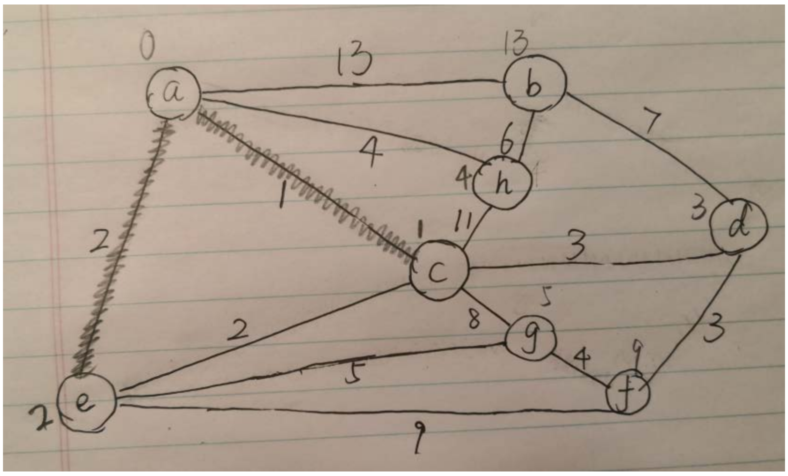
Because a.key is the smallest, we will extract it and update keys for those vertices adjacent to a. Now, A = {}, Q = {b, c, d, e, f, g, h}.



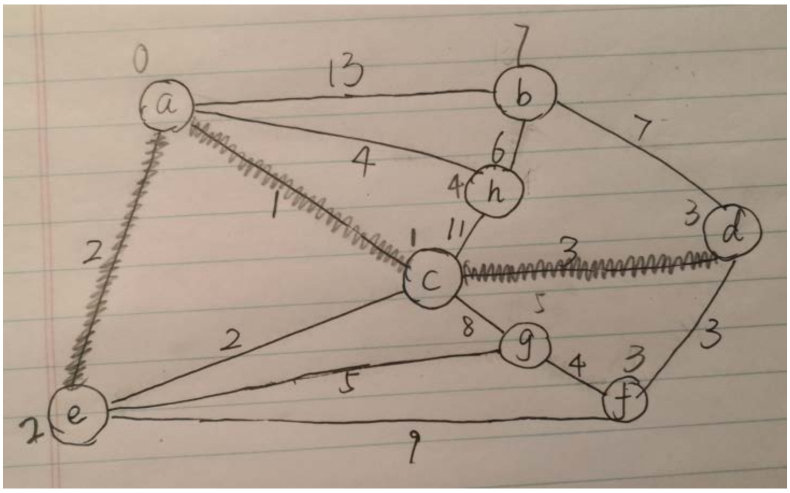
Because c.key is the smallest, we will extract it and add the edge (a, c) to A. Also, updating keys for those vertices adjacent to c. Now, A = {(a, c)}, Q = {b, d, e, f, g, h}.



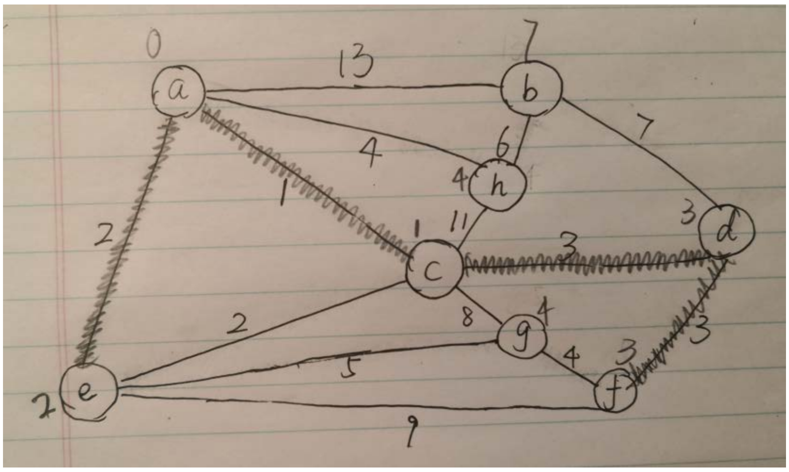

Because e.key is the smallest, we will extract it and add the edge (a, e) to A. Also, updating keys for those vertices adjacent to e. Now, A = {(a, c), (a, e)}, Q = {b, d, f, g, h}.



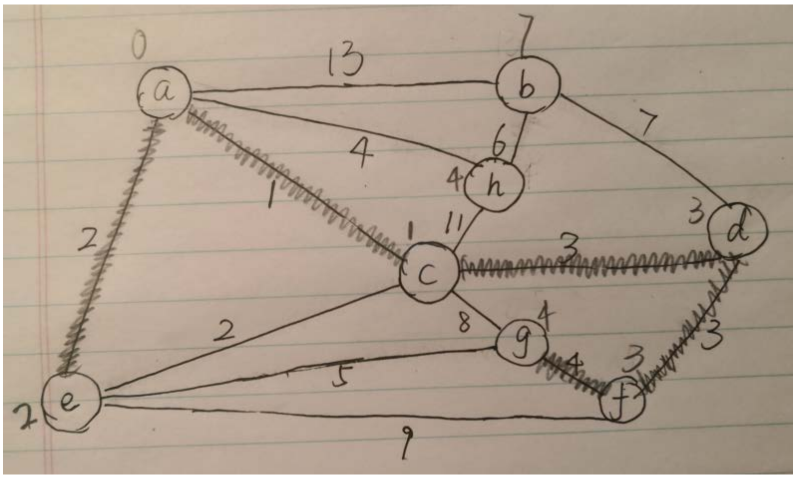

Because d.key is the smallest, we will extract it and add the edge (c, d) to A. Also, updating keys for those vertices adjacent to d. Now, A = {(a, c), (a, e), (c, d)}, Q = {b, f, g, h}.



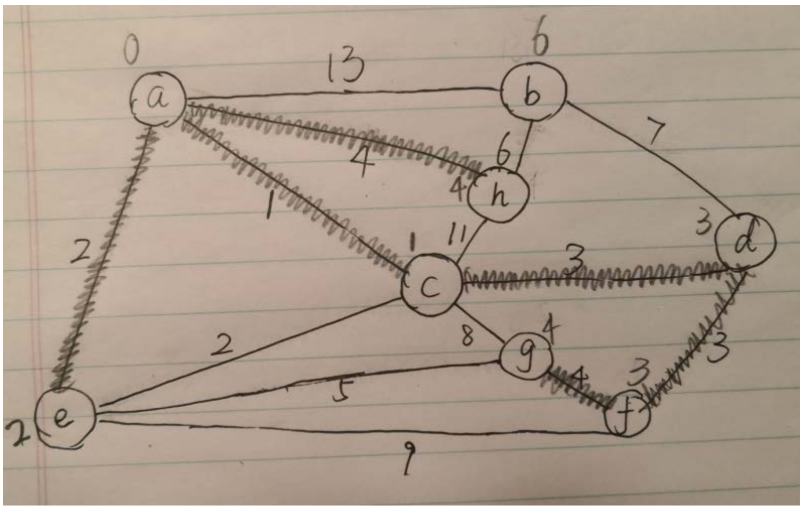

Because f.key is the smallest, we will extract it and add the edge (d, f) to A. Also, updating keys for those vertices adjacent to f. Now, A = {(a, c), (a, e), (c, d), (d, f)}, Q = {b, g, h}.



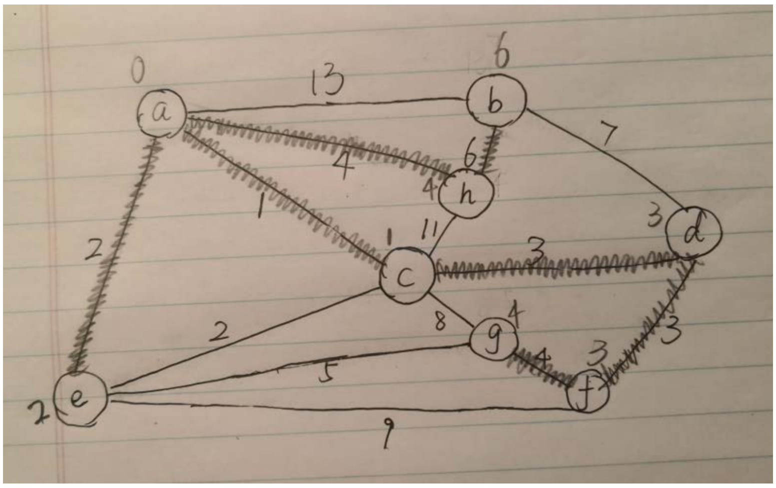

Because h.key and g.key are the smallest, we choose g according to the order of alphabet, we will extract it and add the edge (f, g) to A. Also, updating keys for those vertices adjacent to g. Now, A = {(a, c), (a, e), (c, d), (d, f), (f, g)}, Q = {b, h}.




Because h.key is the smallest, we will extract it and add the edge (a, h) to A. Also, updating keys for those vertices adjacent to h. Now, A = {(a, c), (a, e), (c, d), (d, f), (a, h)}, Q = {b}.



Because b.key is the smallest, we will extract it and add the edge (h, b) to A. Also, updating keys for those vertices adjacent to b. Now, A = {(a, c), (a, e), (c, d), (d, f), (a, h), (b, h)}, Q = {}.




Because Q = ∅, we use Prim algorithm get the MST, which is shown as above. And the order the vertex are removed from Q is a, e, d, f, g, h, b.