TCS-503: Design and Analysis of Algorithms

Graph Algorithms: Minimum Spanning Tree

Unit IV

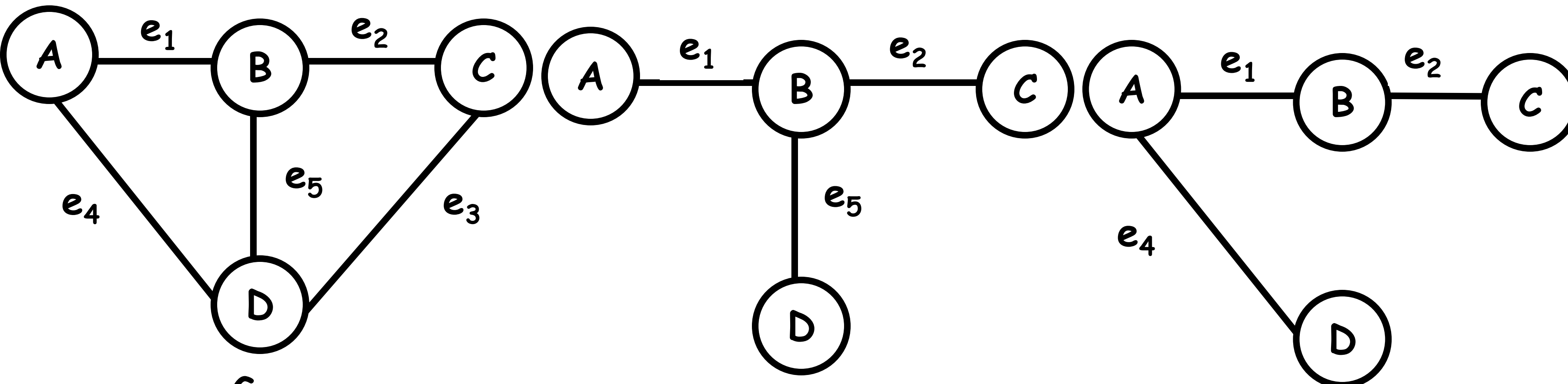
- Graph Algorithms:
  - Elementary Graphs algorithms
  - Minimum Spanning Trees
  - Single-Source Shortest Paths
  - All-Pairs Shortest Paths
  - Maximum Flow and
  - Traveling Salesman Problem

Spanning Tree

A tree T is called a spanning tree of a Graph G if:

T has the same vertices as G and

all the edges of T are contained among the edges of G.



G

If G has n nodes, each spanning tree T must have n-1 edges.

Spanning Tree has no cycles.

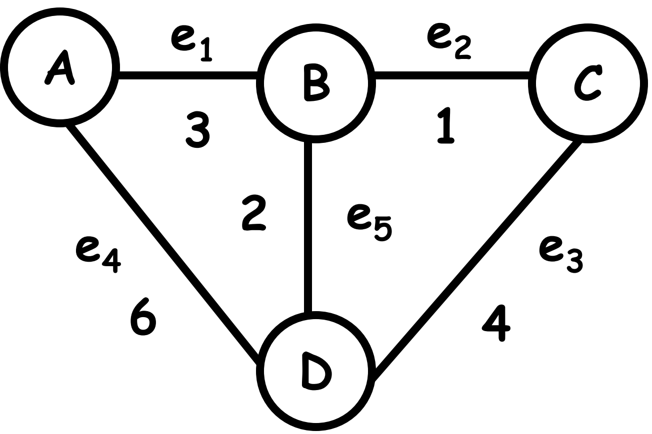
In this example, there are 8 spanning trees.

Minimum Spanning Tree(MST)

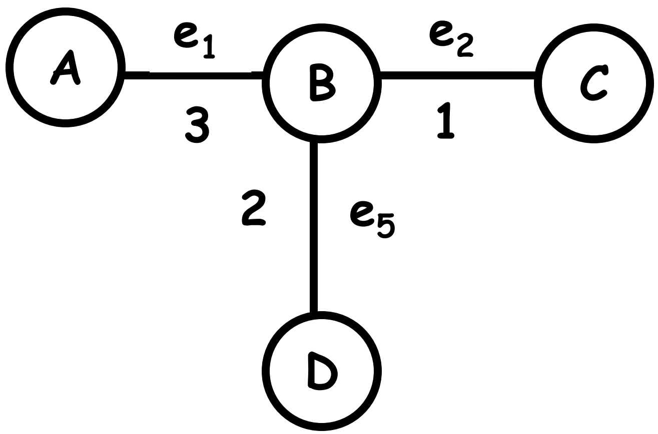
a. k. a

Minimum Weighted Spanning Tree

A minimum spanning tree T of a weighted graph G is a spanning tree of G which has minimum weight among all the spanning trees of G.

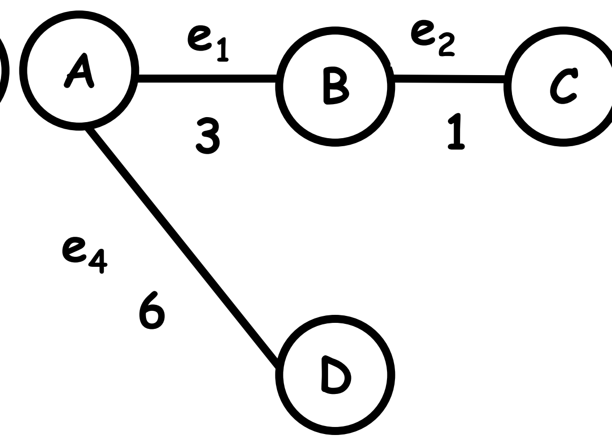


Weighted G



Weight of T_1=

3 + 1 + 2=6



Weight of T_2=

3 + 1 + 6=10

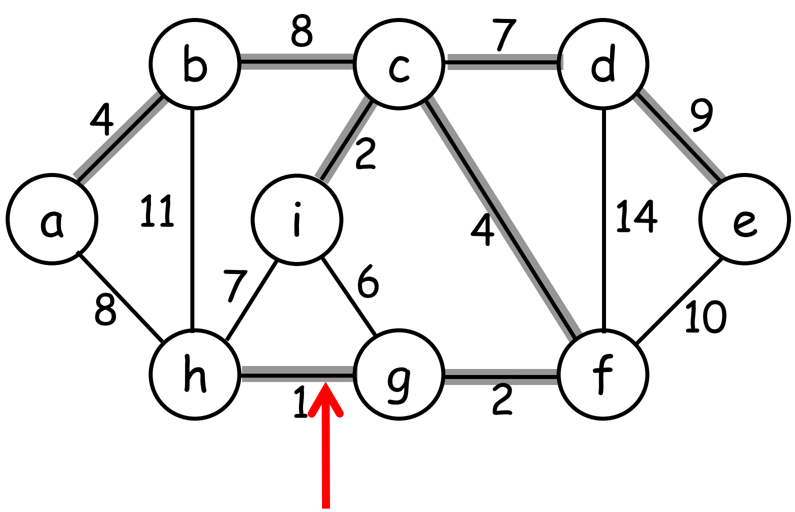
In general, a weighted graph may have more than one MST.

Minimum Spanning Tree(MST)

To Find MST

Kruskal's Algorithm

Starts from minimum weighted edge.



Start from minimum weight edge

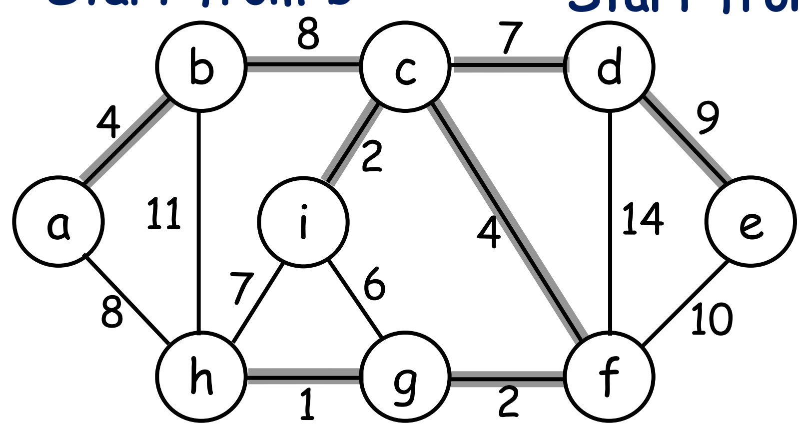
Uses Disjoint sets Data Structure Operations:
MAKE-SET(X), FIND-SET(X) AND UNION(X,Y)

Prim's Algorithm

Starts from arbitrary vertex.

Start from b

Start from d



Start from h

Start from f

Uses a Priority Queue

Minimum Spanning Tree(MST)

Kruskal's algorithm

1. Select the shortest edge in a network.
2. Select the next shortest edge which does not create a cycle.
3. Repeat step 2 until all vertices have been connected.

Prim's algorithm

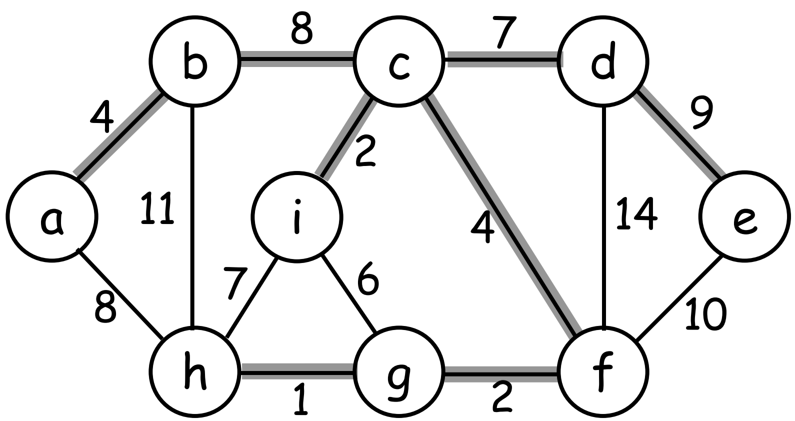
1. Select any vertex.
2. Select the shortest edge connected to that vertex.
3. Select the shortest edge connected to any vertex already connected.
4. Repeat step 3 until all vertices have been connected.

Minimum Spanning Tree(MST)

```
MST_KRUSKAL(G,w)
    A:={}
    for each vertex v in V[G]
          do MAKE_SET(v)
  sort the edges of E by increasing weight w
  for each edge (u,v) in E, in order by increasing weight
          do if FIND_SET(u) !=FIND_SET(v)
                then  A:=AU{(u,v)}
                      UNION(u,v)
  return A
```

Learn DAA : From B K Sharma

Minimum Spanning Tree(MST) Example



Result of Line 1:

A= ∅

Result of Line 2-3:

{a}, {b}, {c}, {d}, {e}, {f}, {g}, {h}, {i}

Result of Line 4:

1: (h, g)

2: (c, i), (g, f)

4: (a, b), (c, f)

6: (i, g)

7:(c, d), (i, h)

8: (a, h), (b, c)

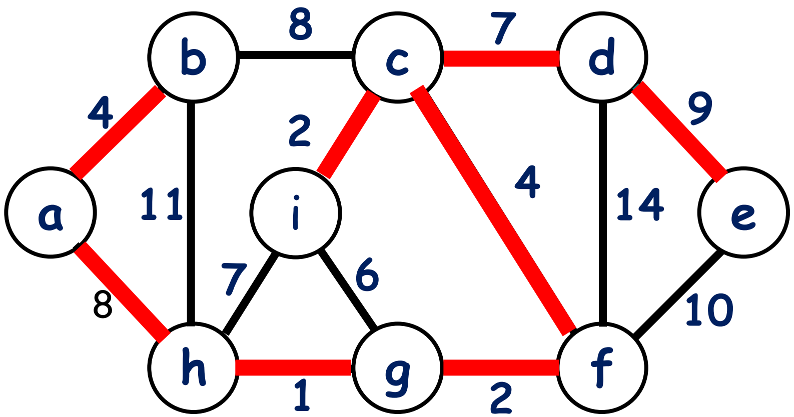
9: (d, e)

10: (e, f)

11: (b, h)

14: (d, f)

w[g, h] +w[c, i] + w[g,f] + w[a,b] + w[c,f] + w[c,d] +w[a,h] +w[d,e]=
1 + 2 + 2 + 4 + 4 + 7 + 8 + 9= 37



Result of Line 1:

A= ∅

Result of Line 2-3:

{a},{b},{c},{d},{e},{f},{g},{h},{i}

Result of Line 4:

1: (h, g)
2: (c, i),(g, f)
4: (a, b),(c, f)
6: (i, g)
7:(c, d),(i, h)
8: (a, h),(b, c)
9: (d, e)
10: (e, f)
11: (b, h)
14: (d, f)

Result of Line 5-8:

1. Add (h, g): {g,h},{a},{b},{c},{d},{e},{f},{i}
2. Add (c, i): {g,h},{c,i},{a},{b},{d},{e},{f}
3. Add (g, f): {g,h,f},{c,i},{a},{b},{d},{e}
4. Add (a, b): {g, h, f}, {c, i}, {a, b},{d},{e}
5. Add (c, f): {g, h, f, c, i}, {a, b}, {d},{e}
6. Ignore (i, g):{g, h, f, c, i}, {a, b}, {d},{e}
7. Add (c, d): {g, h, f, c, i, d}, {a, b}, {e}
8. Ignore (i, h): {g, h, f, c, i, d}, {a, b}, {e}
9. Add (a, h): {g, h, f, c, i, d, a, b}, {e}
10. Ignore (b, c): {g, h, f, c, i, d, a, b}, {e}
11. Add (d, e): {g, h, f, c, i, d, a, b, e}
12. Ignore (e, f): {g, h, f, c, i, d, a, b, e}
13. Ignore (b, h): {g, h, f, c, i, d, a, b, e}
14. Ignore (d, f): {g, h, f, c, i, d, a, b, e}

A= {g, h, f, c, i, d, a, b, e}

Result of Line 9:

{g, h, f, c, i, d, a, b, e}

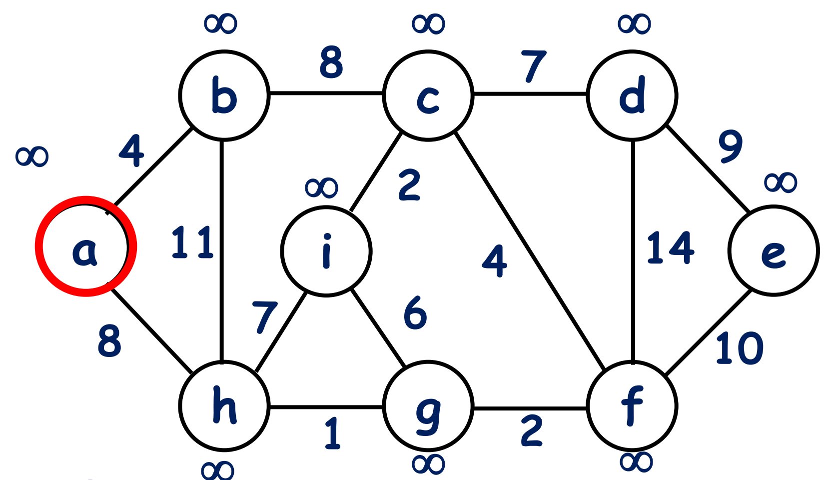
Learn DAA : From B K Sharma

Minimum Spanning Tree(MST)

```
PRIM(G, w, r)
    Q ← ∅
    for each u ∈ V
        do  key[u] ← ∞
                π[u] ← NIL
                INSERT(Q, u)
    V_A = ∅
    DECREASE-KEY(Q, r, 0)
while Q ≠ ∅
        do u ← EXTRACT-MIN(Q)
```

Source : r

0



$Q \leftarrow \varnothing$

Π[a]=NIL, Π[b]=NIL,
Π=[c]=NIL,Π[d]=NIL,
Π[e]=NIL, Π[f]=NIL,
Π[g]=NIL, Π[h]=NIL,
Π[i]=NIL

$Q = \{ \overset{\infty}{a}, \overset{\infty}{b}, \overset{\infty}{c}, \overset{\infty}{d}, \overset{\infty}{e}, \overset{\infty}{f}, \overset{\infty}{g}, \overset{\infty}{h}, \overset{\infty}{i} \}$

$V_A = \varnothing$

$Q = \{ \overset{0}{a}, \overset{\infty}{b}, \overset{\infty}{c}, \overset{\infty}{d}, \overset{\infty}{e}, \overset{\infty}{f}, \overset{\infty}{g}, \overset{\infty}{h}, \overset{\infty}{i} \}$

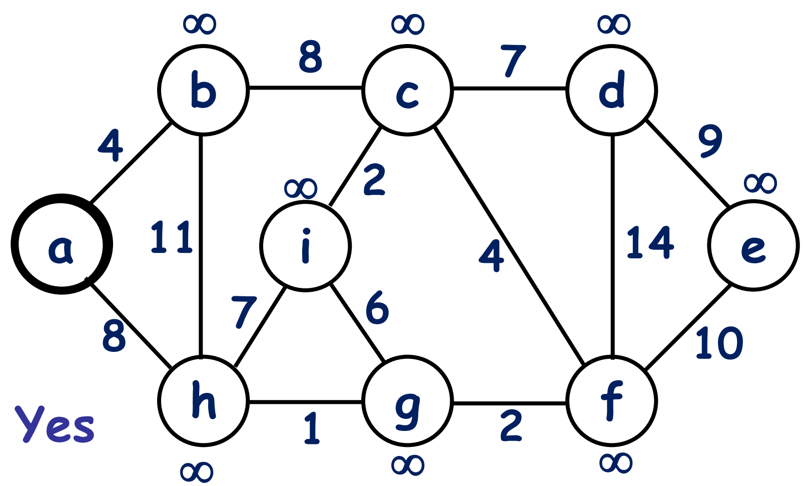
Minimum Spanning Tree(MST)

```
V_A={u}
for each v ∈ Adj[u]
      do if v ∈ Q and w(u, v) < key[v]
          then π[v] ← u
                DECREASE-KEY(Q, v, w(u, v))
```

Minimum Spanning Tree(MST)

u=Extract-MIN(Q) $\Rightarrow$u= a

V_A={a}



11. for each v $\in$ Adj[u]

12. do if v $\in$ Q and w(u, v) < key[v]

b $\in$ Q:Yes and w(a, b) < key[b], i.e. 4 < ∞: Yes

13. then

π[v] ← u

π [b] = a

14. DECREASE-KEY(Q, b, w(a, b))

key [b] = 4

0	∞	∞	∞	∞	∞	∞	∞	∞

Q = { a, b, c, d, e, f, g, h, i }

h $\in$ Q:Yes and w(a, h)< key[h], i.e, 8< ∞: Yes

then

π[v] ← u

π [h] = a

DECREASE-KEY(Q, h, w(a, h))

key [h] = 8

Minimum Spanning Tree(MST)

Extract-MIN(Q) $\Rightarrow$ b

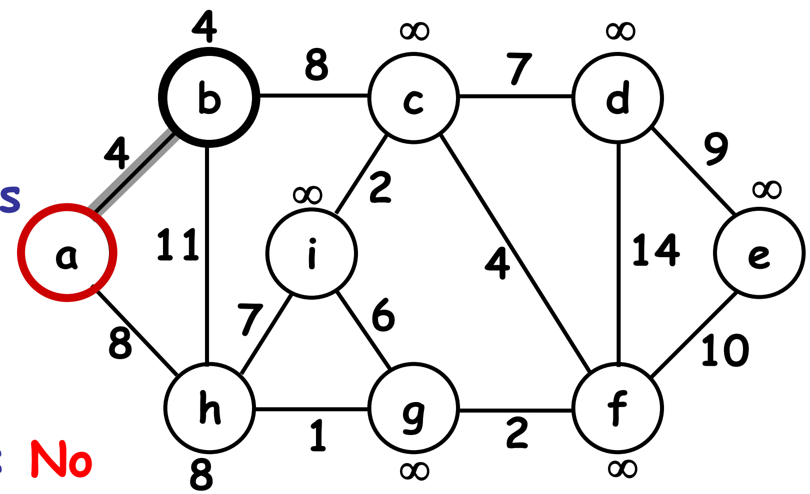
V_A = {a, b}

c $\in$ Q:Yes and w(b, c)< key[c],i.e., 8<∞:Yes

π [h] = a   key [c] = 8

h $\in$ Q:Yes and w(b, h)< key[h], i.e., 11< 8: No

π [h] = a - unchanged   key [h] = 8-unchanged



	4	∞	∞	∞	∞	∞	8	∞
Q = {	b,	c,	d,	e,	f,	g,	h,	i }

Minimum Spanning Tree(MST)

Extract-MIN(Q) $\Rightarrow$ c

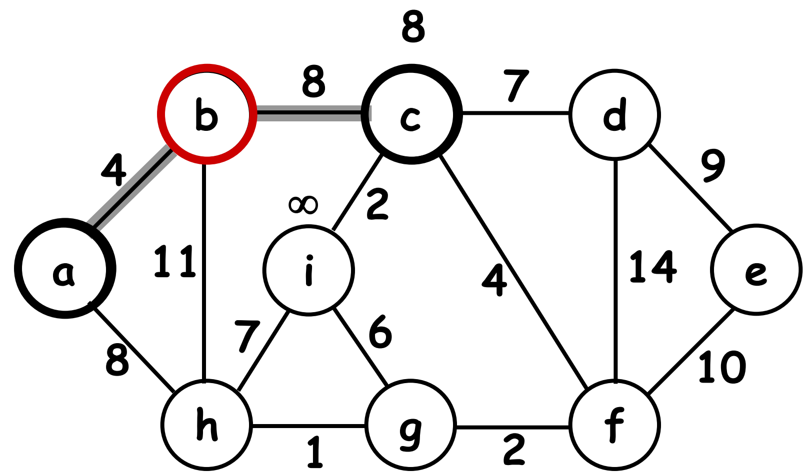
V_A = {a, b, c}

$v \in Q$ and $w(u, v) < key[v]$

π [d] = c key [d] = 7

π [f] = c key [f] = 4

π [i] = c key [i] = 2



8	∞	∞	∞	∞	8	∞
Q = { c,	d,	e,	f,	g,	h,	i }

Learn DAA : From B K Sharma

Minimum Spanning Tree(MST)

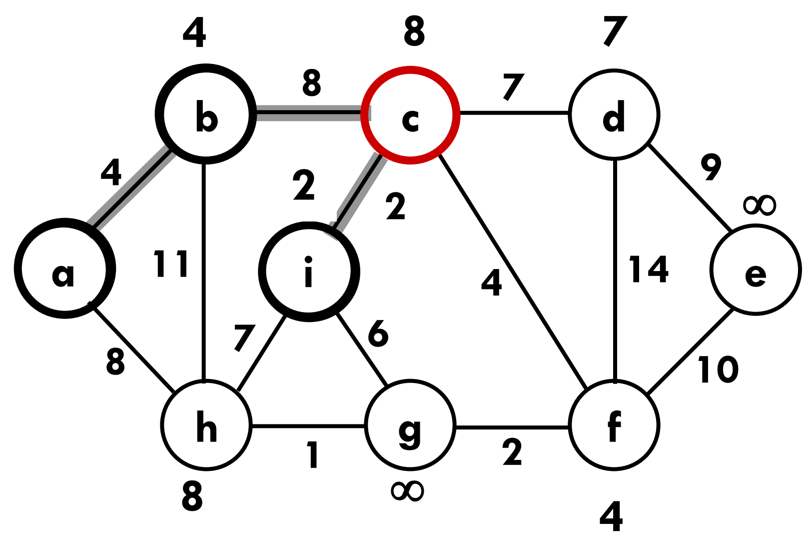
Extract-MIN(Q) $\Rightarrow$ i

V_A = {a, b, c, i }

$v \in Q$ and $w(u, v) < key[v]$

π [h] = i key [h] = 7

π [g] = i key [g] = 6



7 ∞ 4 ∞ 8 2

Q = {d, e, f, g, h, i }

Minimum Spanning Tree(MST)

Extract-MIN(Q) $\Rightarrow$ f

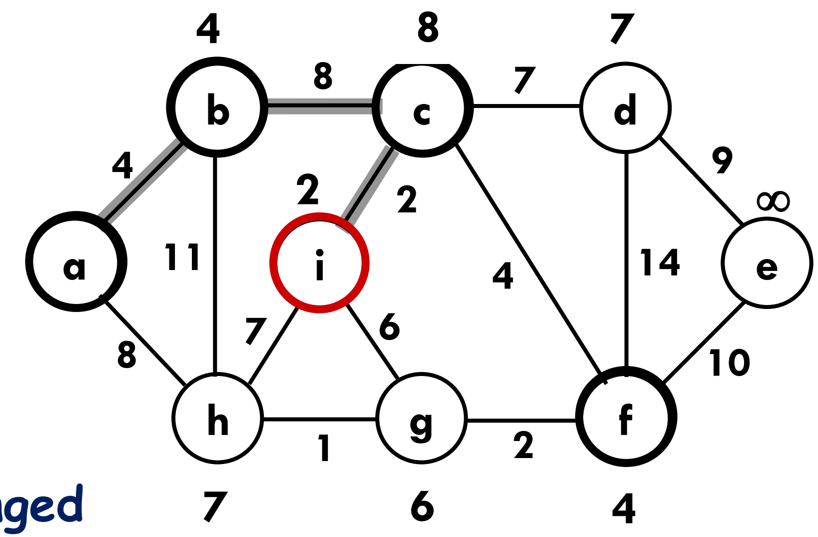
V_A = {a, b, c, i, f}

v $\in$ Q and w(u, v) < key[v]

π [g] = f key [g] = 2

π [e] = f key [e] = 10

π[d]=c-unchanged key[d]=7-unchanged



7 ∞ 4 6 7

Q = {d, e, f, g, h}

Learn DAA : From B K Sharma

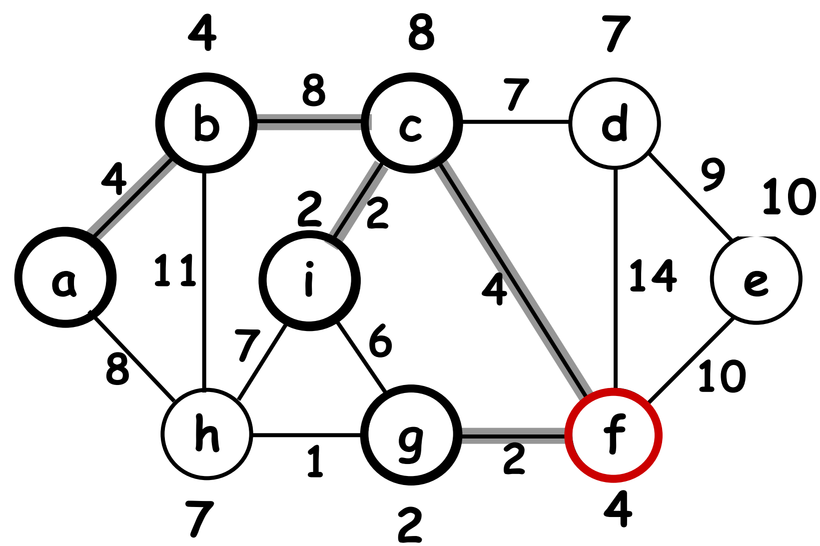
Minimum Spanning Tree(MST)

Extract-MIN(Q) $\Rightarrow$ g

V_A = {a, b, c, i, f, g}

$v \in Q$ and $w(u, v) < key[v]$

π [h] = g   key [h] = 1



7   10   2   7

Q = { d, e, g, h }

Minimum Spanning Tree(MST)

Extract-MIN(Q) $\Rightarrow$ h

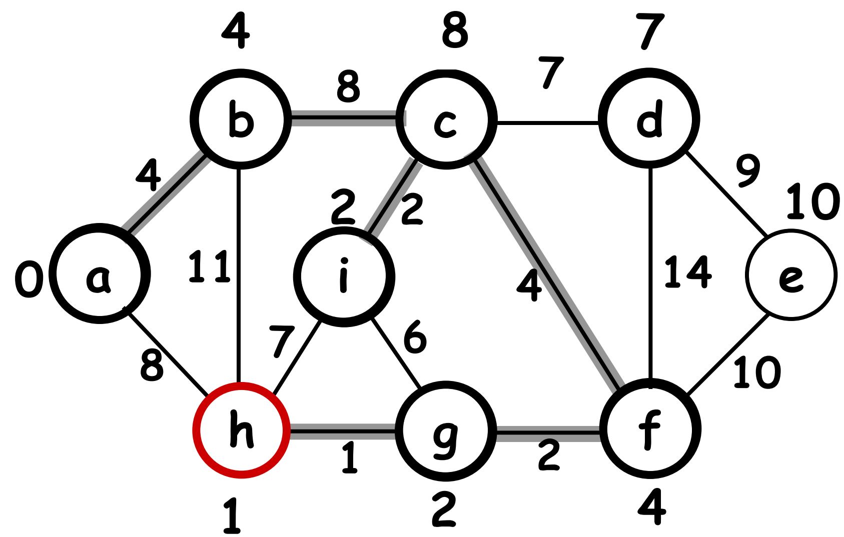
V_A = {a, b, c, i, f, g, h}

v $\in$ Q and w(u, v) < key[v]

Adjacent vertices of h are a, b, i, g.

They do not $\in$ Q

No Key, No π will be changed.



7 10 1

Q = { d, e, h }

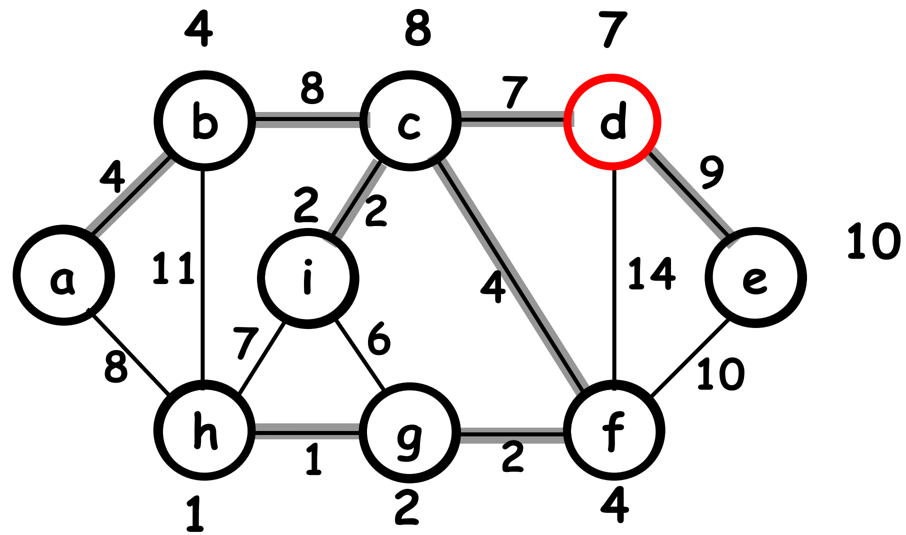
Minimum Spanning Tree(MST)

Extract-MIN(Q) ⇒ d

V_A = {a, b, c, i, f, g, h, d}

v ∈ Q and w(u, v) < key[v]

π[e]=d   key[e]= 9



7 10

Q = { d, e }

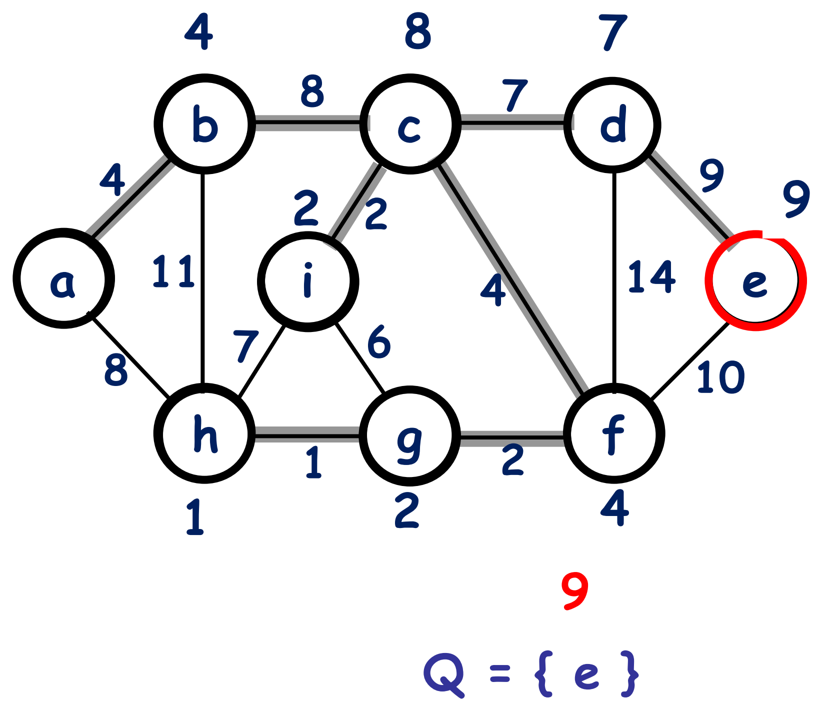
Learn DAA : From B K Sharma

Minimum Spanning Tree(MST)

Extract-MIN(Q) ⇒ e

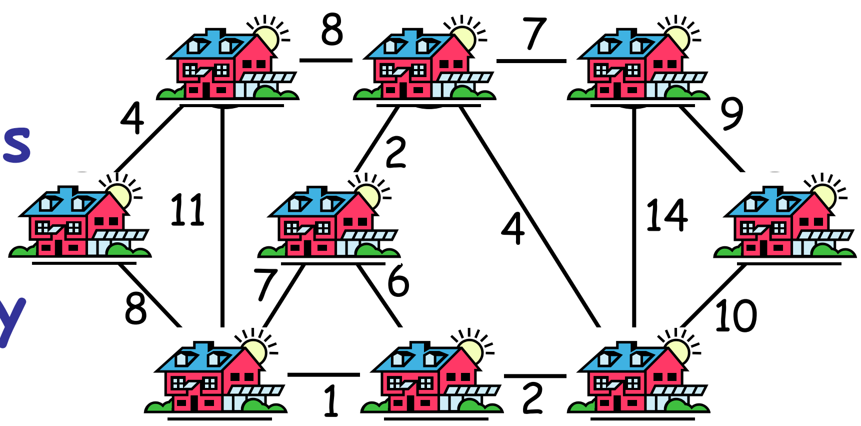
V_A = {a, b, c, i, f, g, h, d, e}

Q = ∅



Minimum Weight: w(a,b) + w(b,c) + w(c,i) +w(c, d) +w(d,e) +w(c,f) +w(f,g) +w(g,h) =4 + 8 +2 +7 +9 +4 +2 +1=37

Minimum Spanning Trees



Problem

- A town has a set of houses and a set of roads
- A road connects 2 and only 2 houses
- A road connecting houses u and v has a repair cost w(u, v)

Goal: Repair enough (and no more) roads such that:

1. Everyone stays connected: can reach every house from all other houses, and
2. Total repair cost is minimum