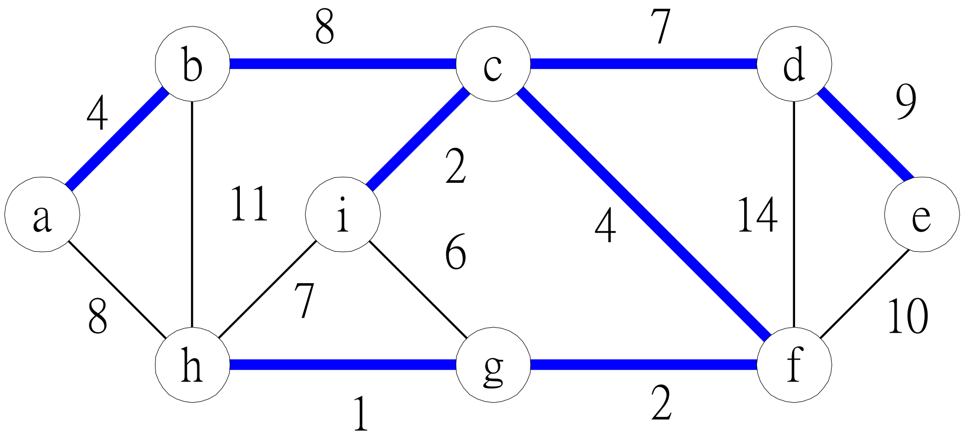
23.Minimum spanning tree

Yu-Shuen Wang, CS, NCTU

Thanks the images from http://alrightchiu.github.io/SecondRound/mu-lu-yan-suan-fa-yu-zi-liao-jie-gou.html

Let G=(V,E) be a connected, undirected graph. For each edge $(u,v) \in E$, we have a weight $w(u,v)$ specifying the cost to connect u and v. We wish to find an acyclic subset $T \subseteq E$ that connects all of the vertices and whose total weight $w(T) = \sum_{(u,v)\in T} w(u,v)$ is minimized. Since T is acyclic and connects all of the vertices, it must form a tree, which we call a ***spanning tree***. We call the problem of determine the tree T the ***minimum spanning tree problem***.



23.1 Growing a minimum spanning tree

GENERIC-MST(*G*, *w*)

1 $A \leftarrow \phi$

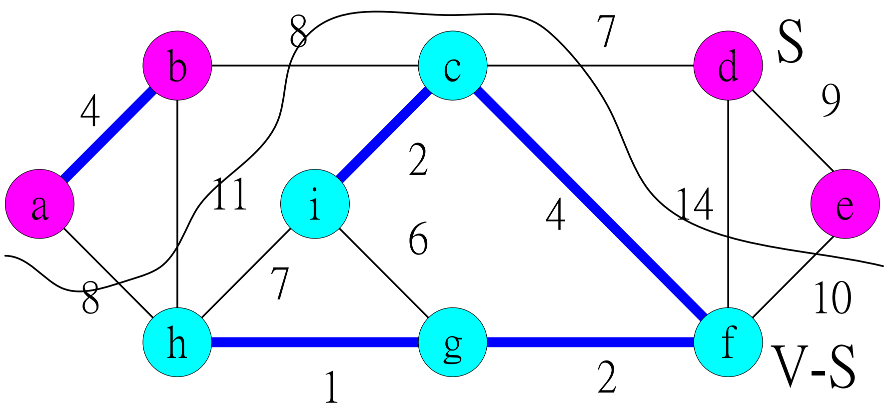
2 **while** *A* does not form a spanning tree

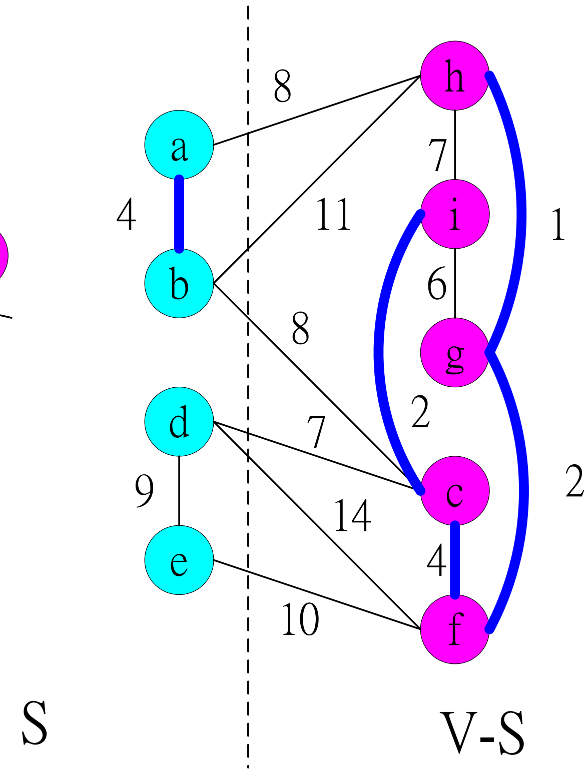
3 **do** find an edge (*u*, *v*) that is ***safe*** for *A*

4 $A \leftarrow A \cup \{(u,v)\}$

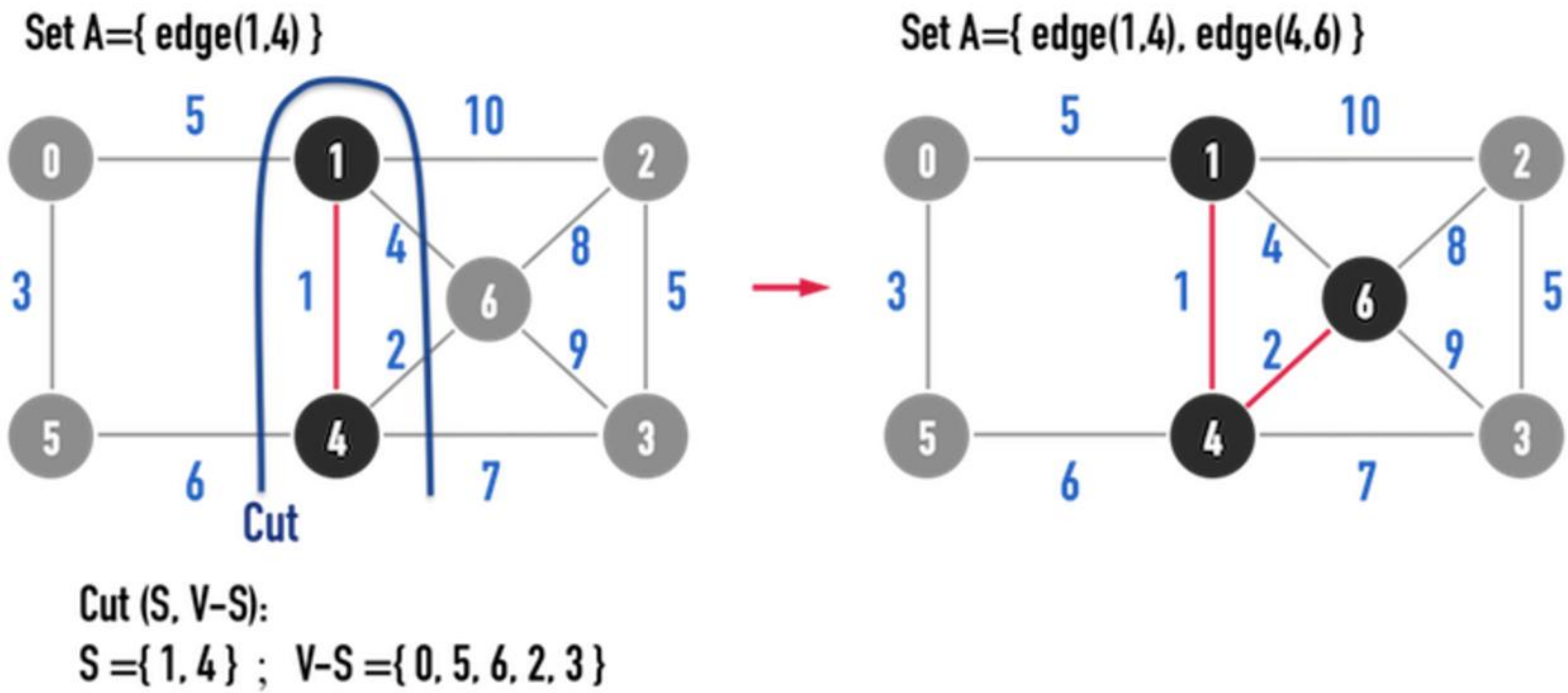
5 **return** *A*

A ***cut*** $(S, V-S)$ of an undirected graph $G = (V,E)$ is a partition of V. We say that an edge $(u,v) \in E$ ***crosses*** the cut (S,V-S) if one of its endpoints is in S and the other is in V-S. We say a cut ***respects*** the set A of edges if no edge in A crosses the cut. An edge is a ***light edge*** crossing a cut if its weight is the minimum of any edge crossing the cut Note that there can be more than one light edge crossing a cut in case of ties.

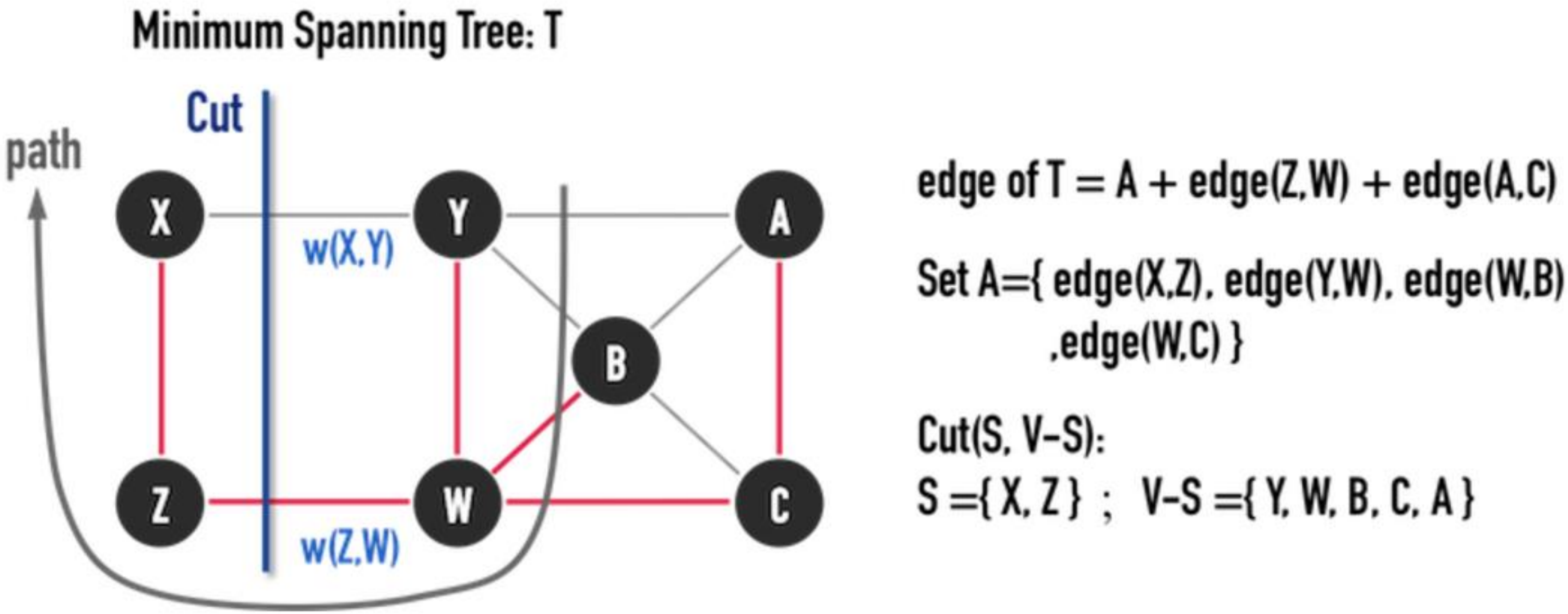

a
b
c
d
e
f
g
h
i
4
8
7
9
11
2
14
4
6
7
8
10
1
2
S
V-S


a
b
d
e
h
i
g
c
f
8
4
11
7
1
6
8
2
7
2
9
14
4
10
S
V-S

Theorem 23.1. Let $G = (V, E)$ be a connected undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V\text{-}S)$ be any cut of G that respects A, and let (u, v) be a light edge crossing $(S, V\text{-}S)$. Then, edge (u, v) is safe for A.



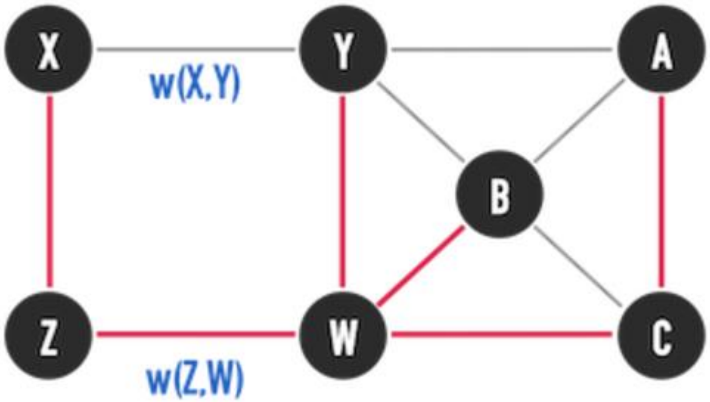
Proof.



Suppose that minimum spanning tree T is composed of edges:
Set A + edge{Z,W} + edge{A,C}, in which Set A is the current state when computing the tree.

Suppose that edge(X,Y) is the light edge among the crossing edges of the Cut. It means weight(X,Y) $\leq$ weight(Z,W)

Minimum Spanning Tree: T

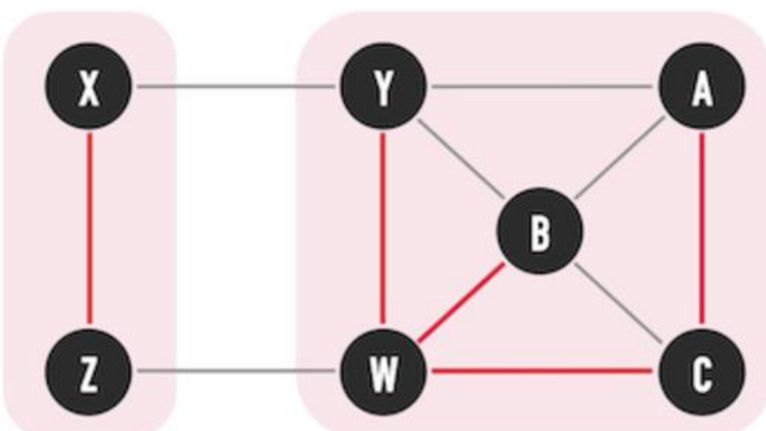


Set A={ edge(X,Z), edge(Y,W), edge(W,B) ,edge(W,C) }

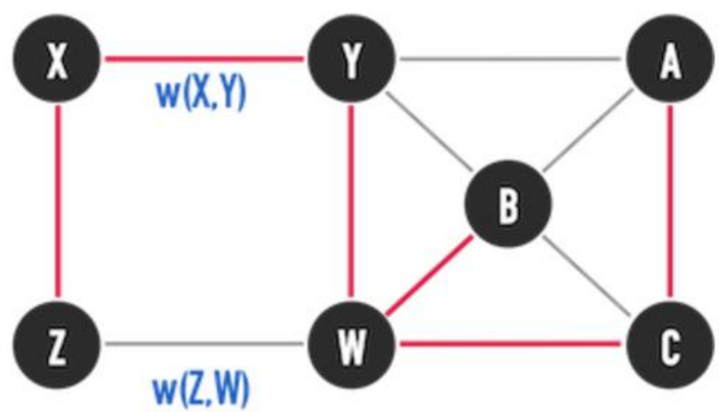
edge of T = A + edge(Z,W) + edge(A,C)

edge of T' = A + edge(X,Y) + edge(A,C)

Two Connected Components

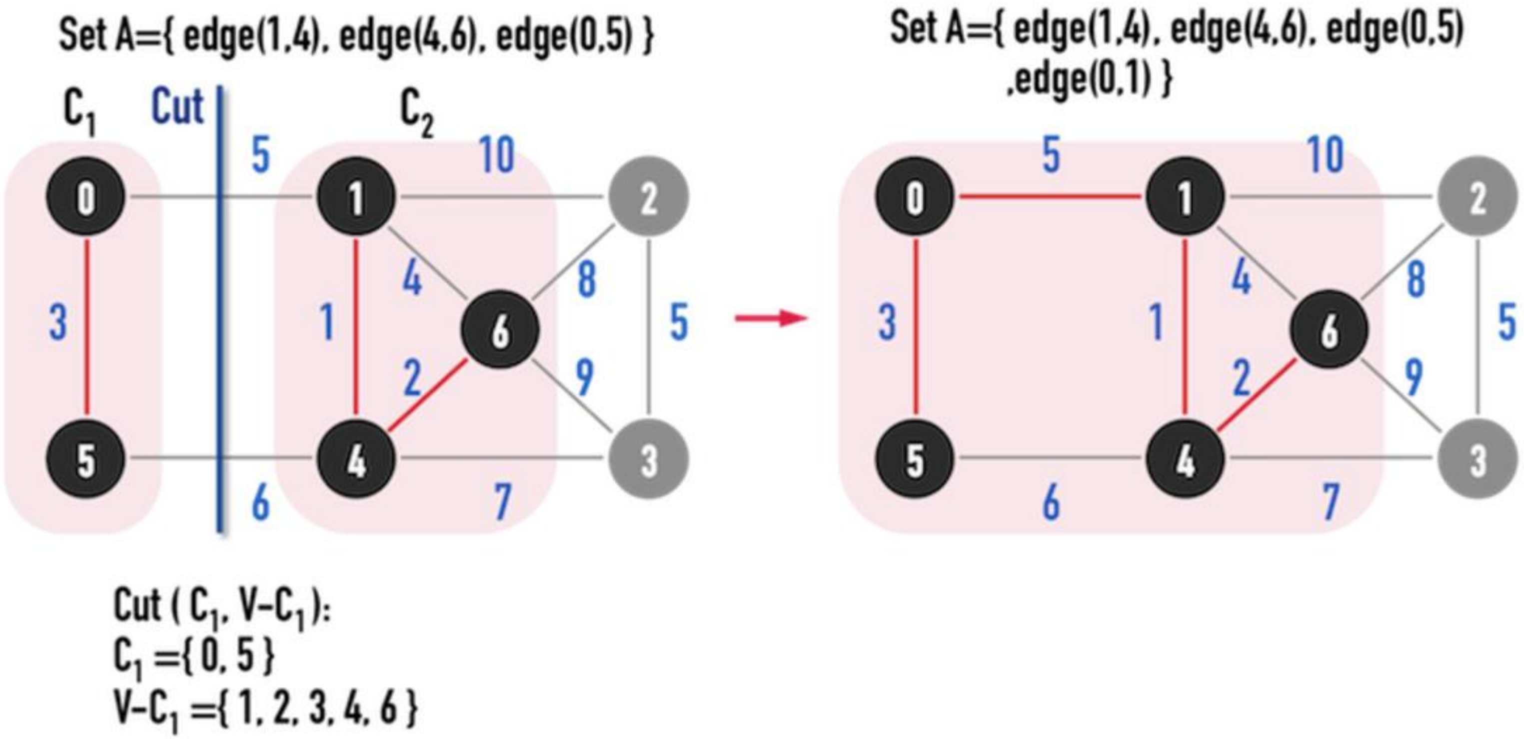


Minimum Spanning Tree: T'

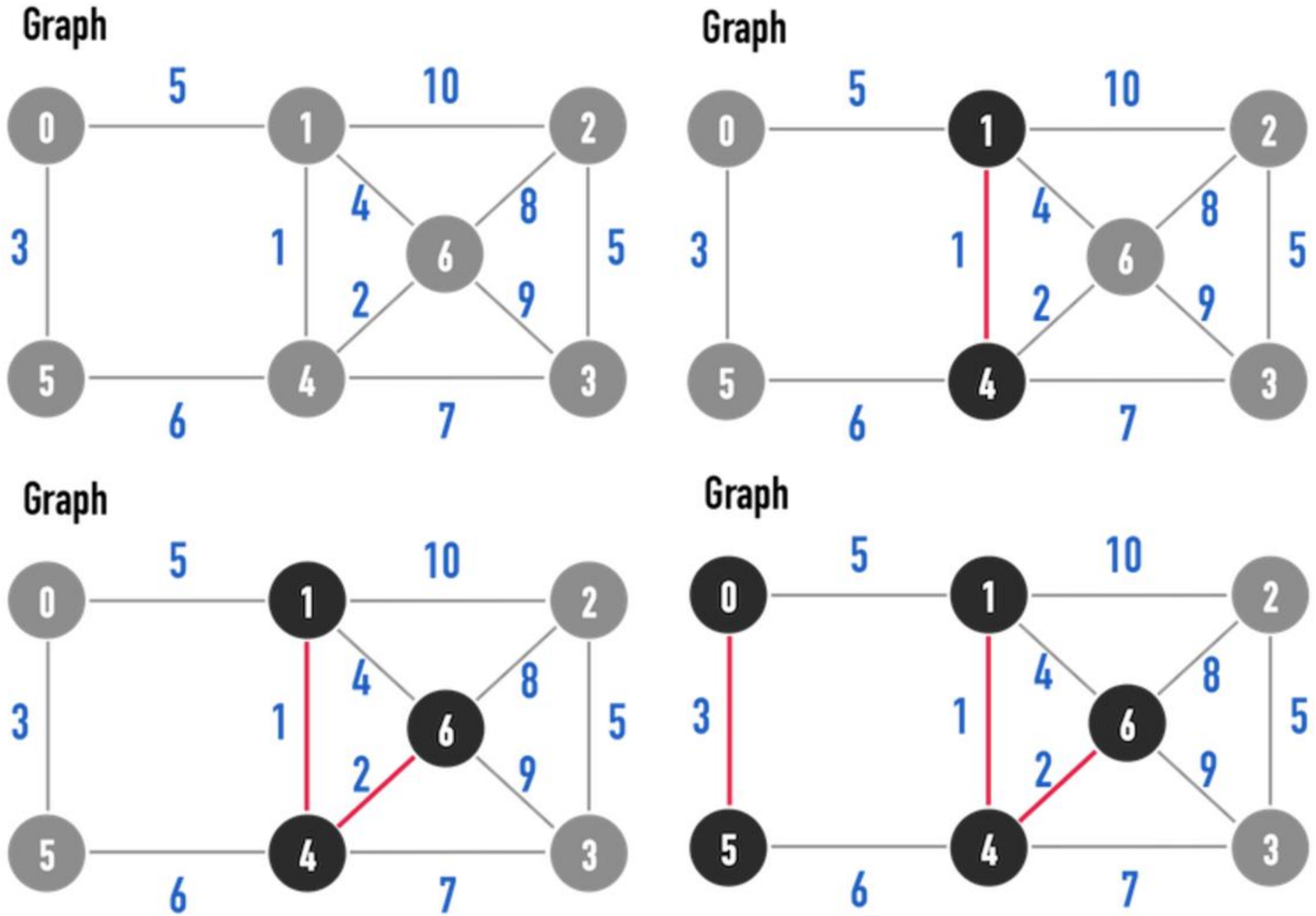


$$weight(T') \leq weight(T) - weight(Z,W) + weight(X,Y) \ ;$$
$$weight(T') \leq weight(T) \ ;$$

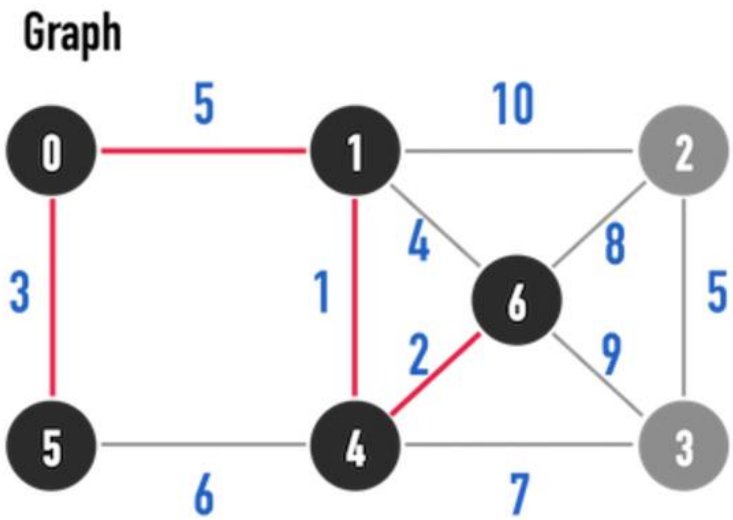
Corollary 23.2. Let $G = (V, E)$ be a connected, undirected graph with a real-valued weighted function w defined on E. Let A be a subset of E that is induced in some minimum spanning tree for G, and let C be a connected component (tree) in the forest $G_A = (V, A)$. If (u,v) is a light edge connecting C to some other component G_A, then (u, v) is safe for A.

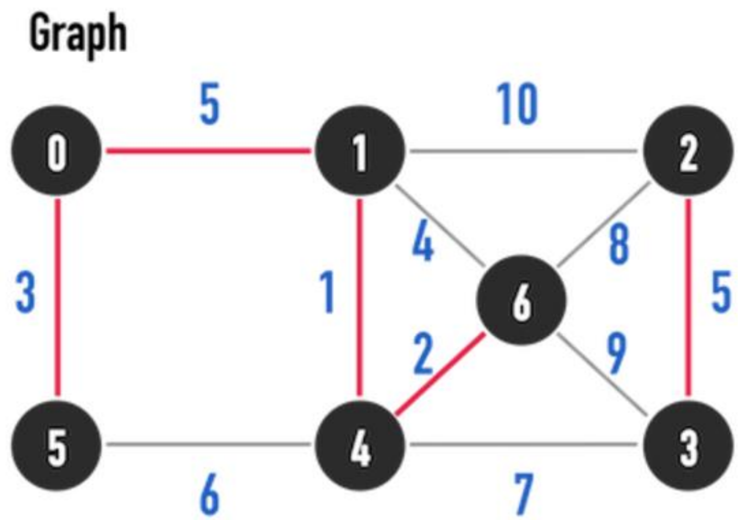


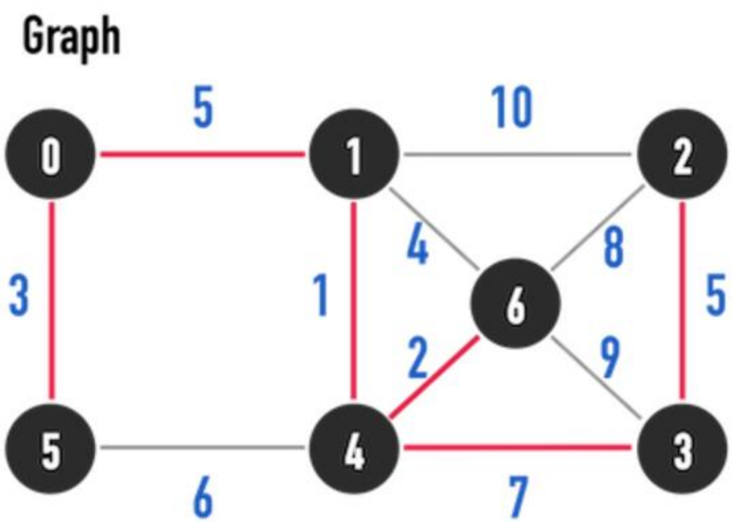
Kruskal’s algorithm



Kruskal’s algorithm







Kruskal's algorithm

MST_KRUSKAL(G, w)

$A \leftarrow \phi$

for each vertex $v \in V[G]$

    do MAKE-SET(v)

sort the edge of E by nondecreasing weight w

for each edge $(u,v) \in E$, in order by nondecreasing weight

    do if FIND_SET(u)≠FIND_SET(v)

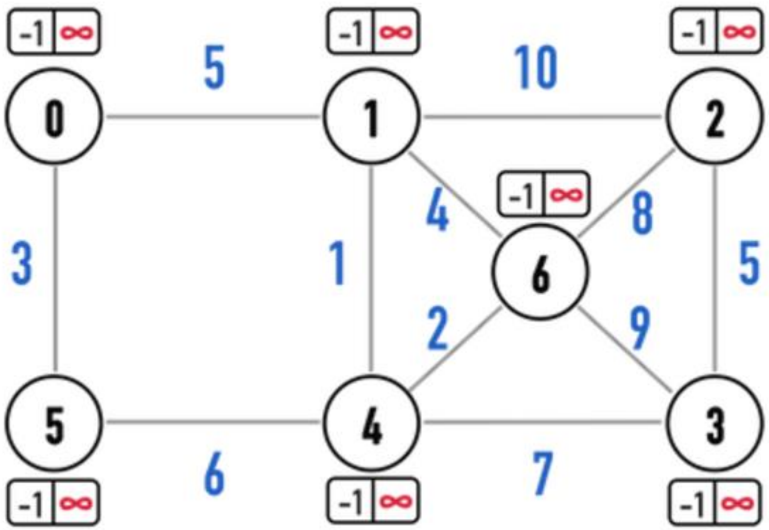
        then $A \leftarrow A \cup \{(u,v)\}$

            UNION(u, v)

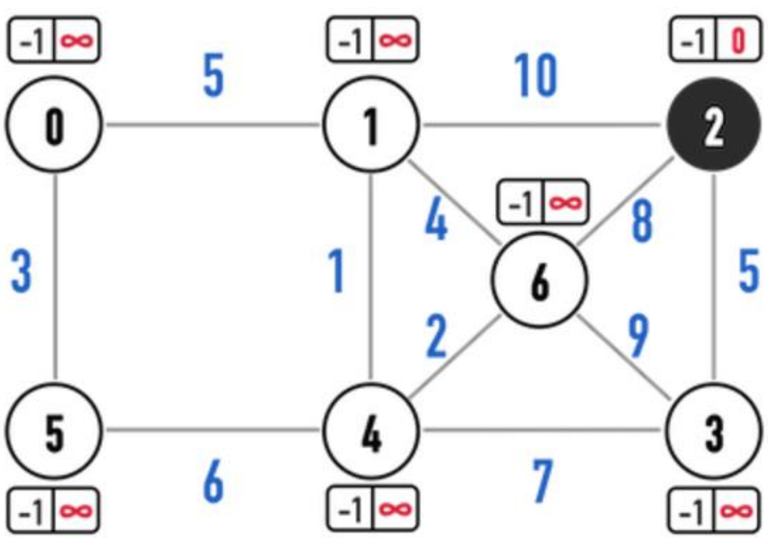
return A

$\boxed{\text{Complexity } O(E \log E)}$

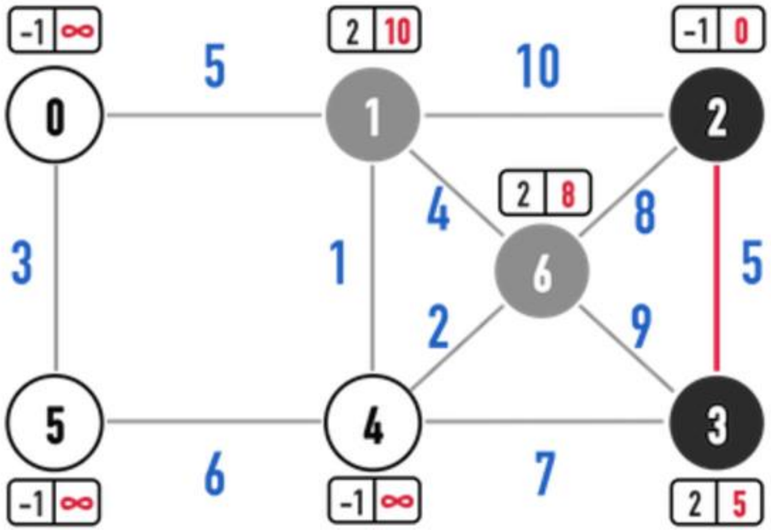
Prim's algorithm



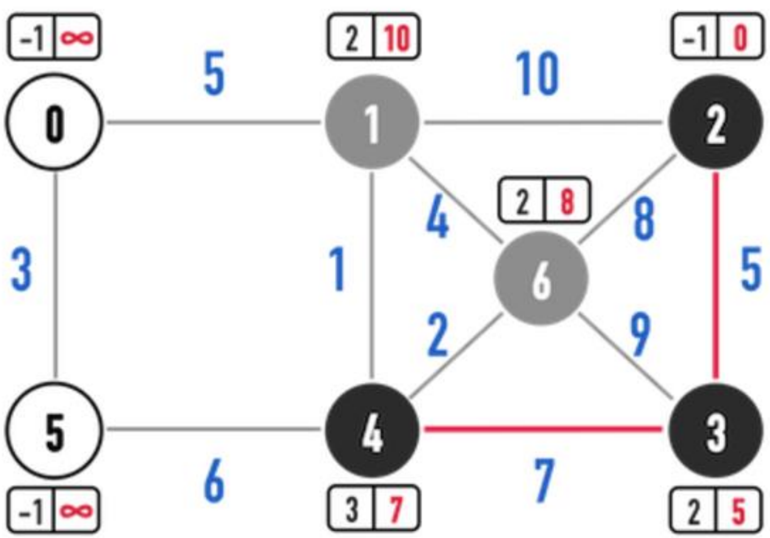
	0	1	2	3	4	5	6
predecessor [ ]	-1	-1	-1	-1	-1	-1	-1
key [ ]	∞	∞	∞	∞	∞	∞	∞
visited [ ]	0	0	0	0	0	0	0



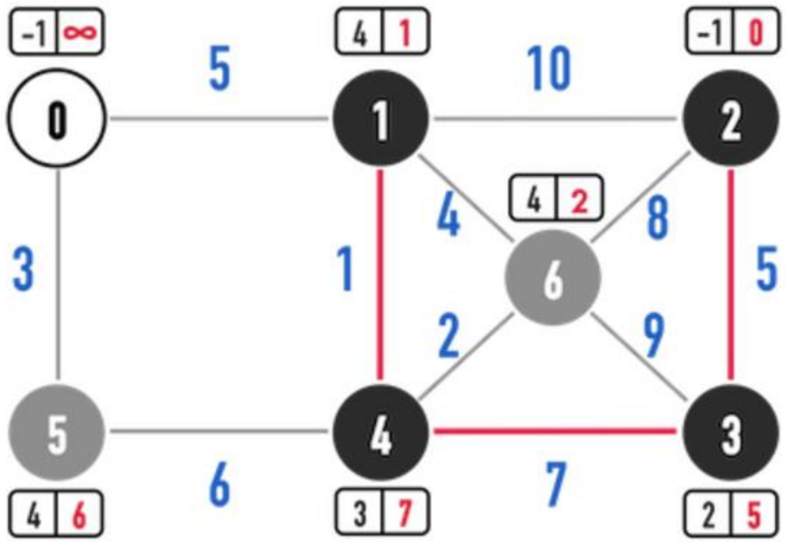
	0	1	2	3	4	5	6
predecessor [ ]	-1	-1	-1	-1	-1	-1	-1
key [ ]	∞	∞	0	∞	∞	∞	∞
visited [ ]	0	0	1	0	0	0	0



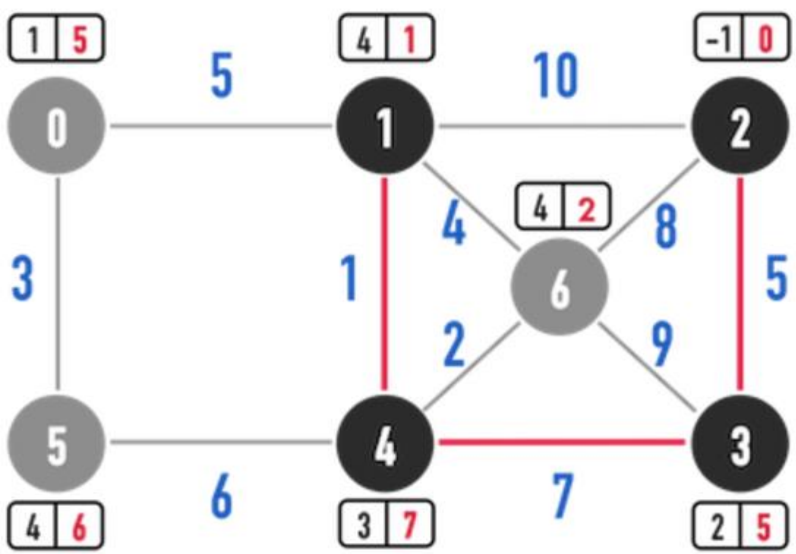
	0	1	2	3	4	5	6
predecessor [ ]	-1	2	-1	2	-1	-1	2
key [ ]	∞	10	0	5	∞	∞	8
visited [ ]	0	0	1	1	0	0	0



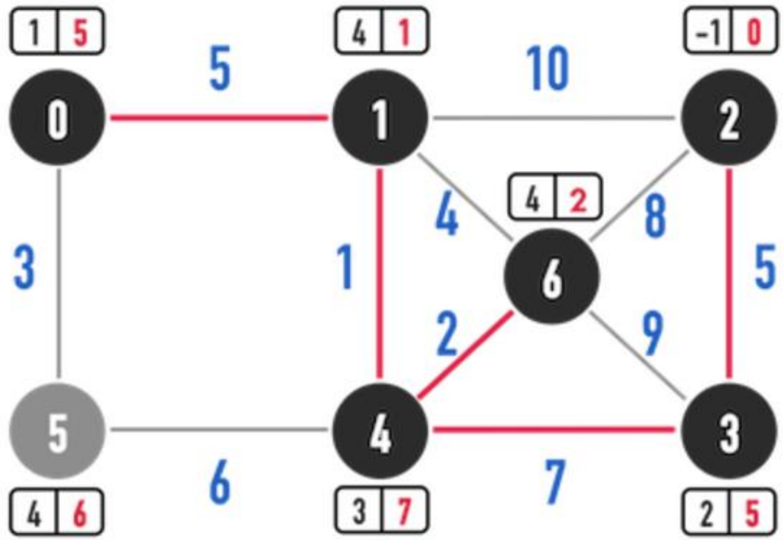
	0	1	2	3	4	5	6
predecessor [ ]	-1	2	-1	2	3	-1	2
key [ ]	∞	10	0	5	7	∞	8
visited [ ]	0	0	1	1	1	0	0



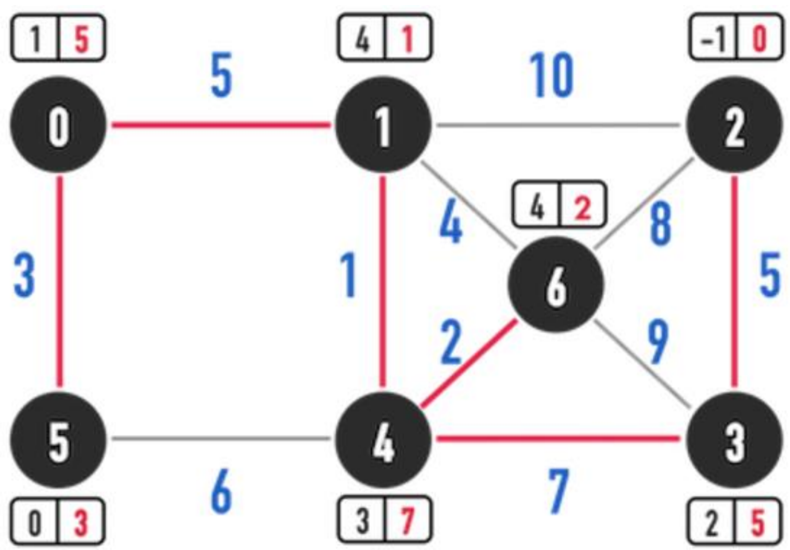
	0	1	2	3	4	5	6
predecessor [ ]	-1	4	-1	2	3	4	4
key [ ]	∞	1	0	5	7	6	2
visited [ ]	0	1	1	1	1	0	0



	0	1	2	3	4	5	6
predecessor [ ]	1	4	-1	2	3	4	4
key [ ]	5	1	0	5	7	6	2
visited [ ]	0	1	1	1	1	0	0



	0	1	2	3	4	5	6
predecessor [ ]	1	4	-1	2	3	4	4
key [ ]	5	1	0	5	7	6	2
visited [ ]	1	1	1	1	1	0	1



	0	1	2	3	4	5	6
predecessor [ ]	1	4	-1	2	3	0	4
key [ ]	5	1	0	5	7	3	2
visited [ ]	1	1	1	1	1	1	1

Prim's algorithm

MST_PRIM(G, w, r)

$Q \leftarrow V[G]$

for each $u \in Q$

  do $key[u] \leftarrow \infty$

$key[r] \leftarrow 0$

$\pi[r] \leftarrow NIL$

while $Q \neq \phi$

  do $u \leftarrow ECTRACT\_MIN(Q)$

    for each $v \in Adj[u]$

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

        then $\pi[v] \leftarrow u$

          $key[v] \leftarrow w(u,v)$

Complexity:
O(V log V + E log V), or
O(E + V log V)