## Formation of Minimum Spanning Tree

*Description:*

Assume that there has one Connected Undirected Graph G = (V, E) and the weight function w: E->R, we hope to find one Minimum Spanning Tree of Graph G. Two Greedy algorithms in this Chapter would be discussed to help solve this question, but the way to use Greedy Algorithm is totally different.

*General Strategy:*

In each time, the General Strategy would generate one edge of the Minimum Spanning Tree, and during the procedure to apply this strategy, the edge collection A has to be managed:

*Before each cycle, the definition A is the Sub-Collection of Minimum Spanning Tree.*

In each step, what we need to do is to choose one edge (u, v) from Graph, and add it into collection A, and make sure that the collection A does not disobey the Loop Invariant*(循环不变式)*. Here, it means that *A Union (u, v)* is the *Sub - Collection of Minimum Spanning Tree*.

Since we can add this edge safely into the Sub-Collection A, and never disobey the Loop Invariant, under such condition, we can call the edge (u, v) as the safe edge.

*Pseudo Code:*

Generic\_MST(G, w) {

A = Empty Set;

while A is not the Minimum Spanning Tree

find one edge (u, v) in all Graph Edges which is safe for collection A

A = A Union edge (u, v);

return A;

}

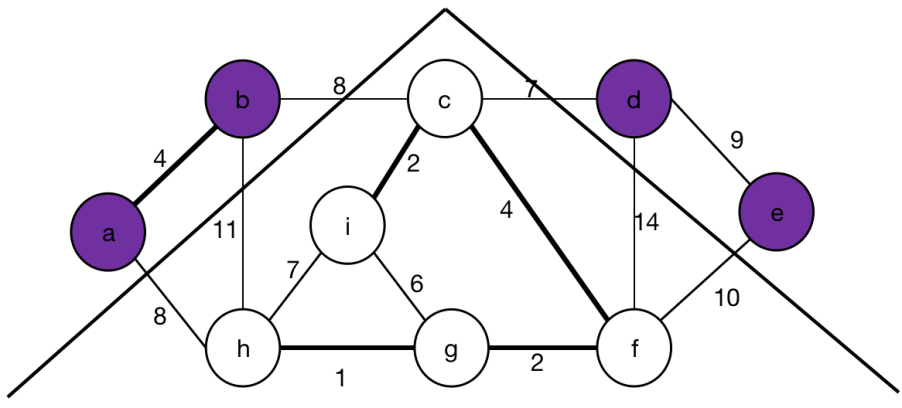
*Conception - Safety Edge:*

*Undirected Graph G = (V, E)* has one division (S, V - S) among Collection V. If there has an Edge (u, v) which belongs to the Collection E. One Point u belongs to Collection S, while another point v belongs to Collection V - S, so the Edge (u, v) divide (S, V - S).

However, if there has no such Edge in Collection A, then such division (D, A - D) respect Collection A.

Among all such Edge in Collection A, the least weight Edge is called *Light Weight Edge*. *Attention that, the Light Weight Edge will not be unique.* If one edge has the least weight which satisfies the specific condition, then this Edge can be called as *one Light Weight Edge*.

*For Example:*

**

In Graph above, there have two Divisions:

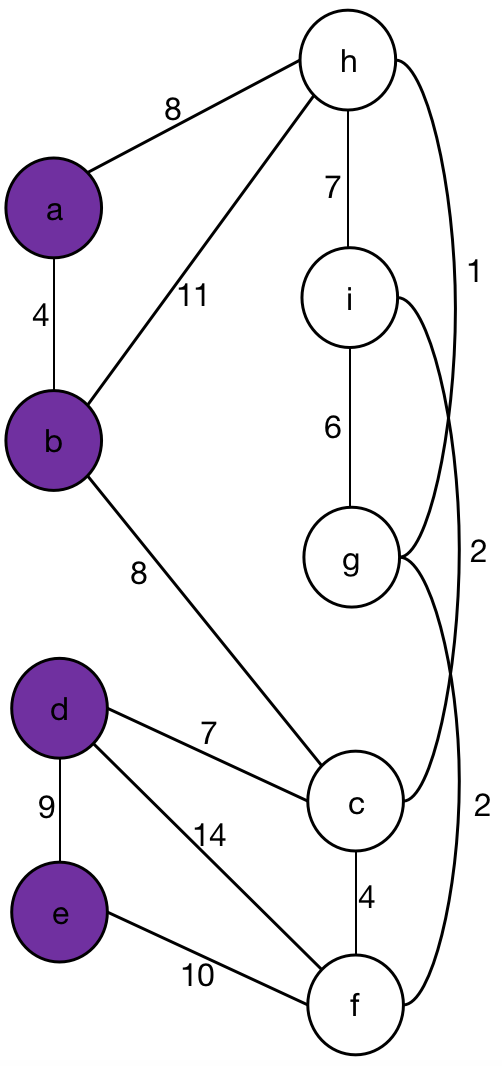
*S = { V, E }, V = { a, b, d, e }, E = { a<->b, d<->e }*

*V - S = { V - S, E }, V - S = { c, i, h, g, f }, E = { c<->i, c<->f, i<->h, i<->g, h<->g, g<->g }*

In Graph below, there also have two Divisions:

*S = { V, E }, V = { a, b, d, e }, E = { a<->b, d<->e }*

*V - S = { V - S, E }, V - S = { c, i, f, g, h }, E = { c<->i, c<->f, i<->h, i<->g, h<->g, g<->g }*



*Rule:*