**Minimum Spanning Tree: Dafuq is it?**

MST is a term used for describing undirected acyclic graphs (aka graphs that don’t specify direction, and no routes taken can lead back to the start point). This means that the routes (see the T H I C C lines to the right) will form a tree. The gimmick with these is that the edges (the connections between vertices aka. The lettered points shown to the right) have a weight/cost. These weights can represent such things as the amount of wire needed to connect some set of pins, or the time between a set of actions occurring. The second part of the gimmick is that the total cost of all the edges of the tree must equal some specific amount, which is decided on a case by case basis. In the image to the upper right, this total amount is 37.

For a more technical explanation:

Undirected Acyclic ,

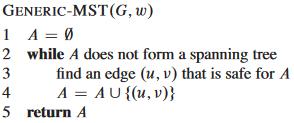
,

,

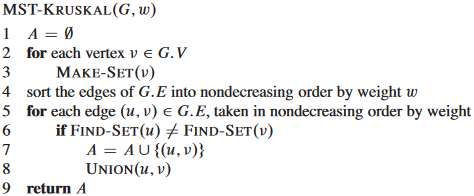
,

**MST Build Functions**

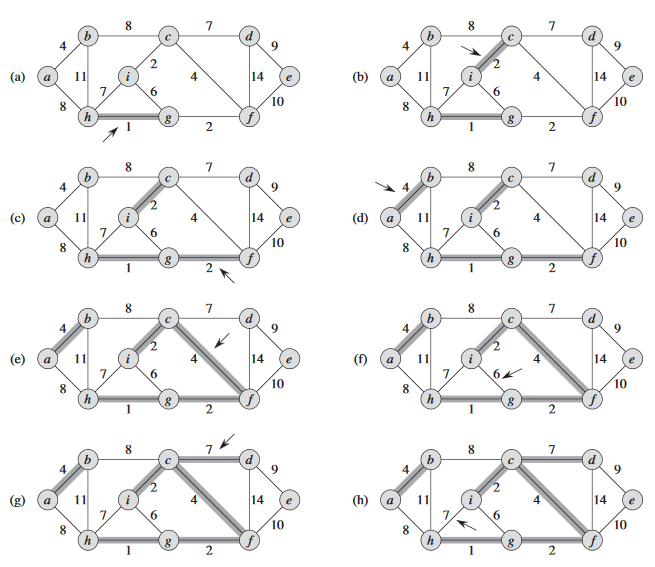
*General*

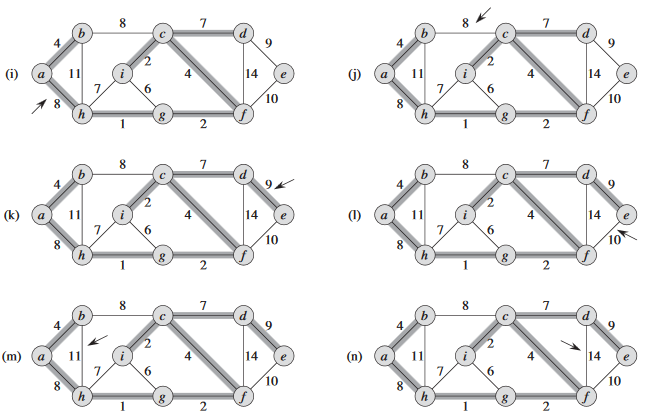


*Kruskal*



FIND-SET(x) finds whichever set a vertex is part of, and returns an identifier for that set. In the above function, it is used to check if two vertices of an edge are part of the same set, where a set represents a single tree.

Below is an example of how the algorithm is run, just for reference:



*Note: Maybe investigate Prim’s algorithm for MST?*