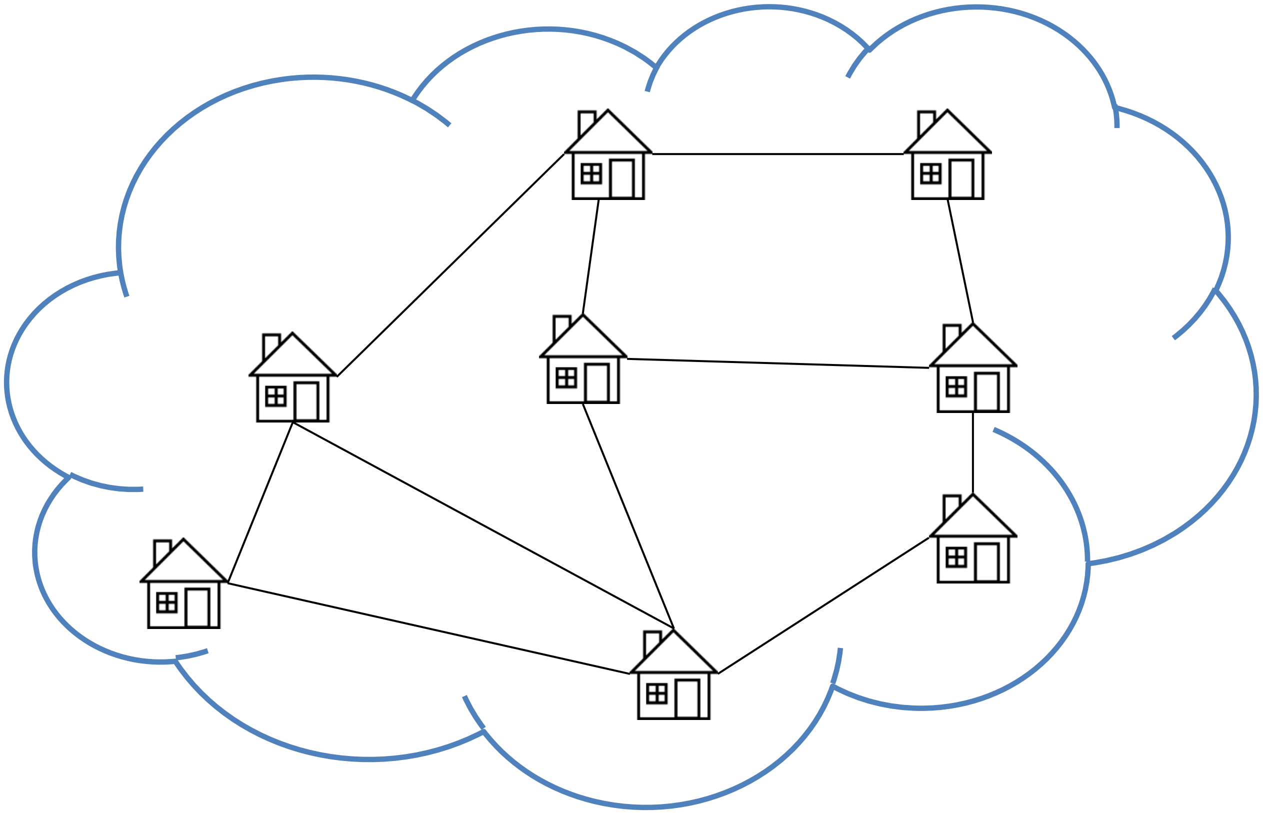
Spanning Trees

Outline

- Spanning tree
  - Minimum spanning tree
  - Prim's algorithm
  - Kruskal's algorithm

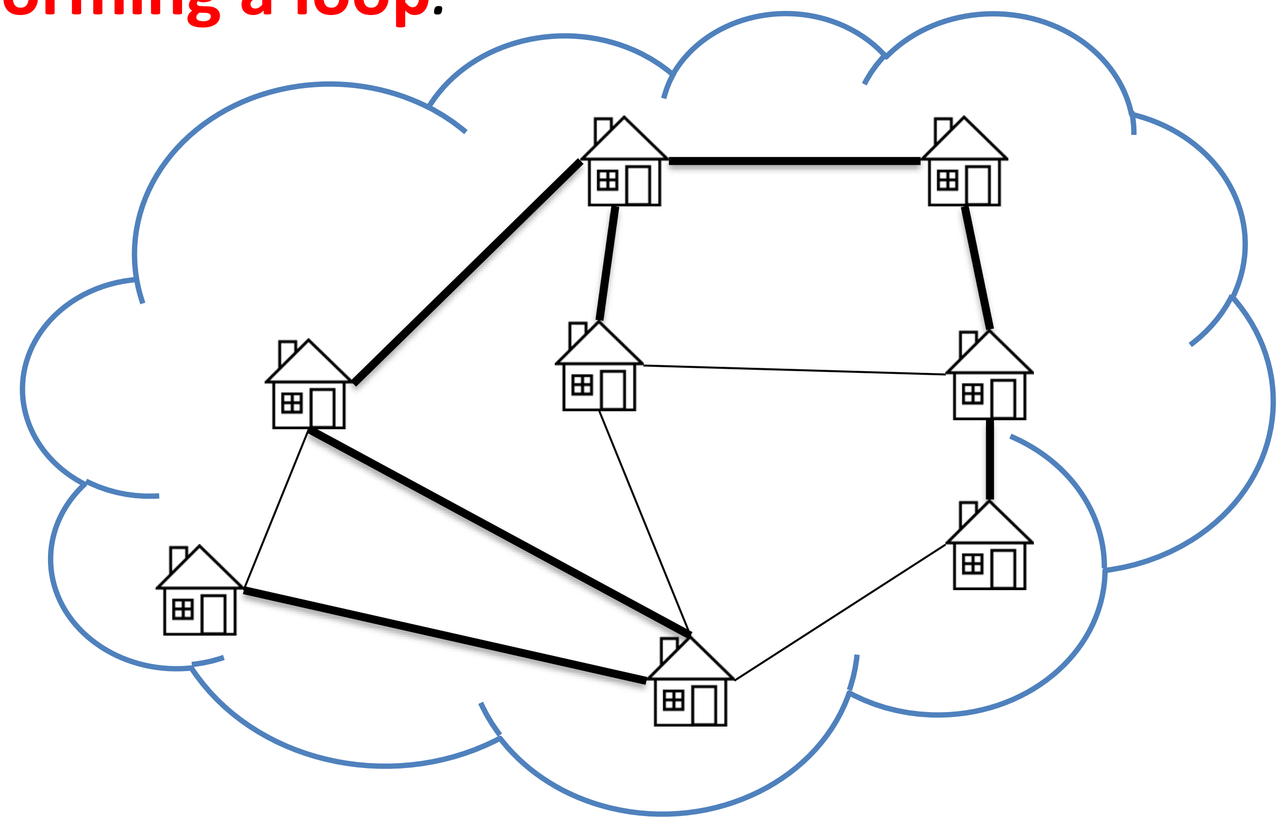
Spanning Trees

- Imagine the scenario where a cable company has to run a cable that will connect **ALL** the houses to internet.



Spanning Trees

- For the company to be efficient, it would like to find a unique path that would go through all the houses, **without forming a loop**.



Definition

- Given a connected, undirected graph $G=(V,E)$, a tree that uses the edges, E, from G and contains all of the vertices, V, is called a **spanning tree of G**.

- Since we are dealing with a tree, the set of vertices and edges must be acyclic.

- If there are N nodes in the graph, there will be exactly N - 1 edges in the tree. The graph may have more than N edges.

- The trees are also unrooted and unordered, unlike other trees we've been working with.

Definition

- Given a connected, undirected graph $\mathrm{G}=(\mathrm{V},\mathrm{E})$, a tree that uses the edges, E, from G and contains all of the vertices, V, is called a **spanning tree of G**.

- Since we are dealing with a tree, the set of vertices and edges must be acyclic.

- If there are N nodes in the graph, there will be exactly N - 1 edges in the tree. The graph may have more than N edges.

- The trees are also unrooted and unordered, unlike other trees we've been working with.

Definition

- Given a connected, undirected graph $G=(V,E)$, a tree that uses the edges, E, from G and contains all of the vertices, V, is called a **spanning tree of G**.

- Since we are dealing with a tree, the set of vertices and edges must be acyclic.

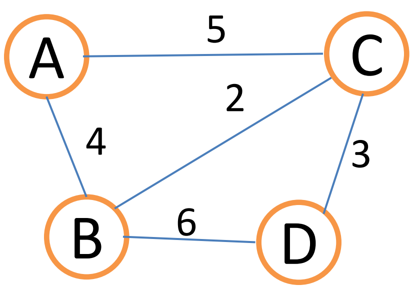
- If there are N nodes in the graph, there will be exactly edges in the tree. The graph may have more than N edges.

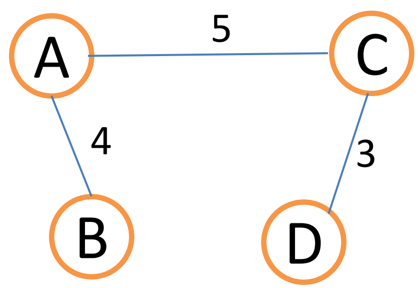
- The trees are also unrooted and unordered, unlike other trees we've been working with.

Definition

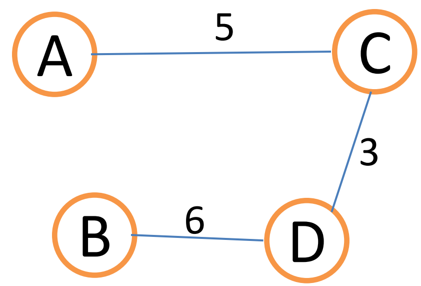
- Given a connected, undirected graph $G=(V,E)$, a tree that uses the edges, E, from G and contains all of the vertices, V, is called a **spanning tree of G**.
- Since we are dealing with a tree, the set of vertices and edges must be acyclic.
- If there are N nodes in the graph, there will be exactly N - 1 edges in the tree. The graph may have more than N edges.
- The trees are also unrooted and unordered, unlike other trees we've been working with.

Spanning Tree Example

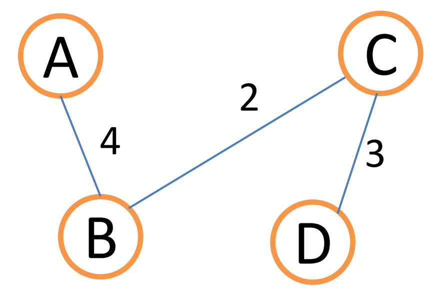




Cost: 12



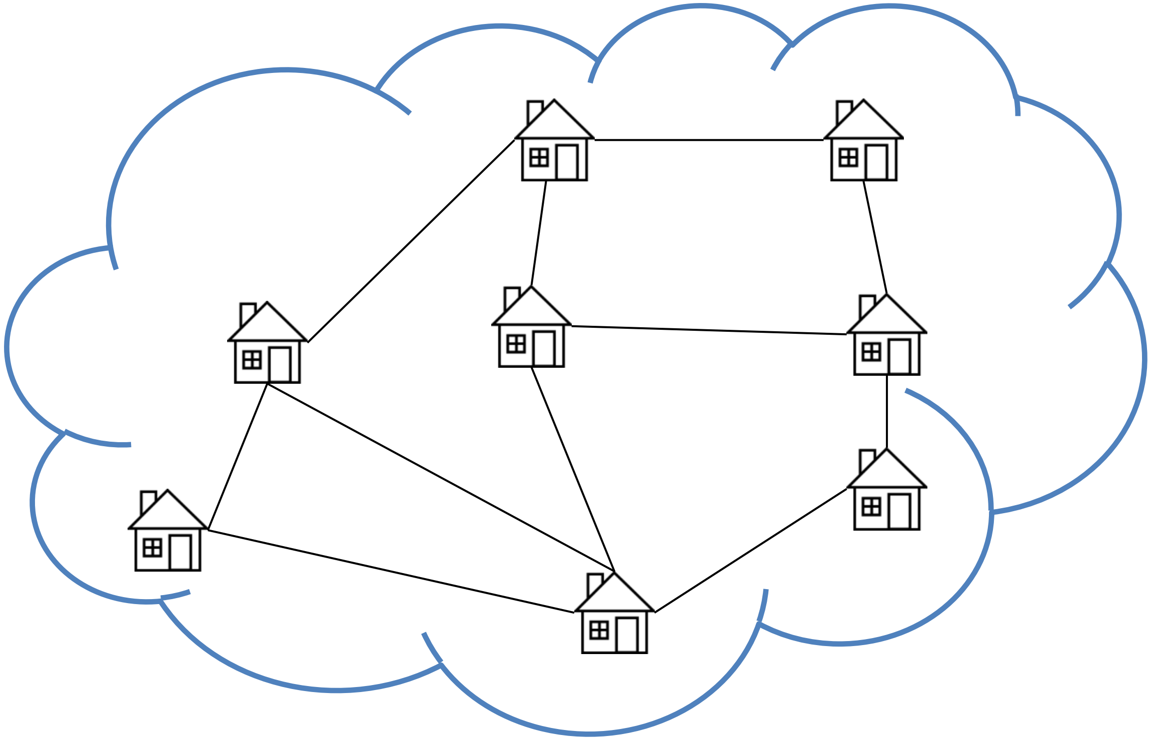
Cost: 14



Cost: 9

Spanning Trees

- Connecting a house to another has a fixed price that varies due to distance and also due to external circumstances.
- Now, the cable company would like to find the spanning tree that would lead to the minimum cost.



Minimum Spanning Tree

- If the cost is minimized, the tree is a minimal spanning tree. More accurately, it might be called a minimum-weighted spanning tree.
- Used in many situations, especially networking and communications: (Spanning Tree Protocol)
  - May have many routes between computers, but you just want one set that connects everyone in the cheapest way.
  - Cheap could mean actual monetary cost or could mean "fastest" (in which case you may want to maximize the cost.)
- What is the number of edges in a MST?

Prim’s Algorithm

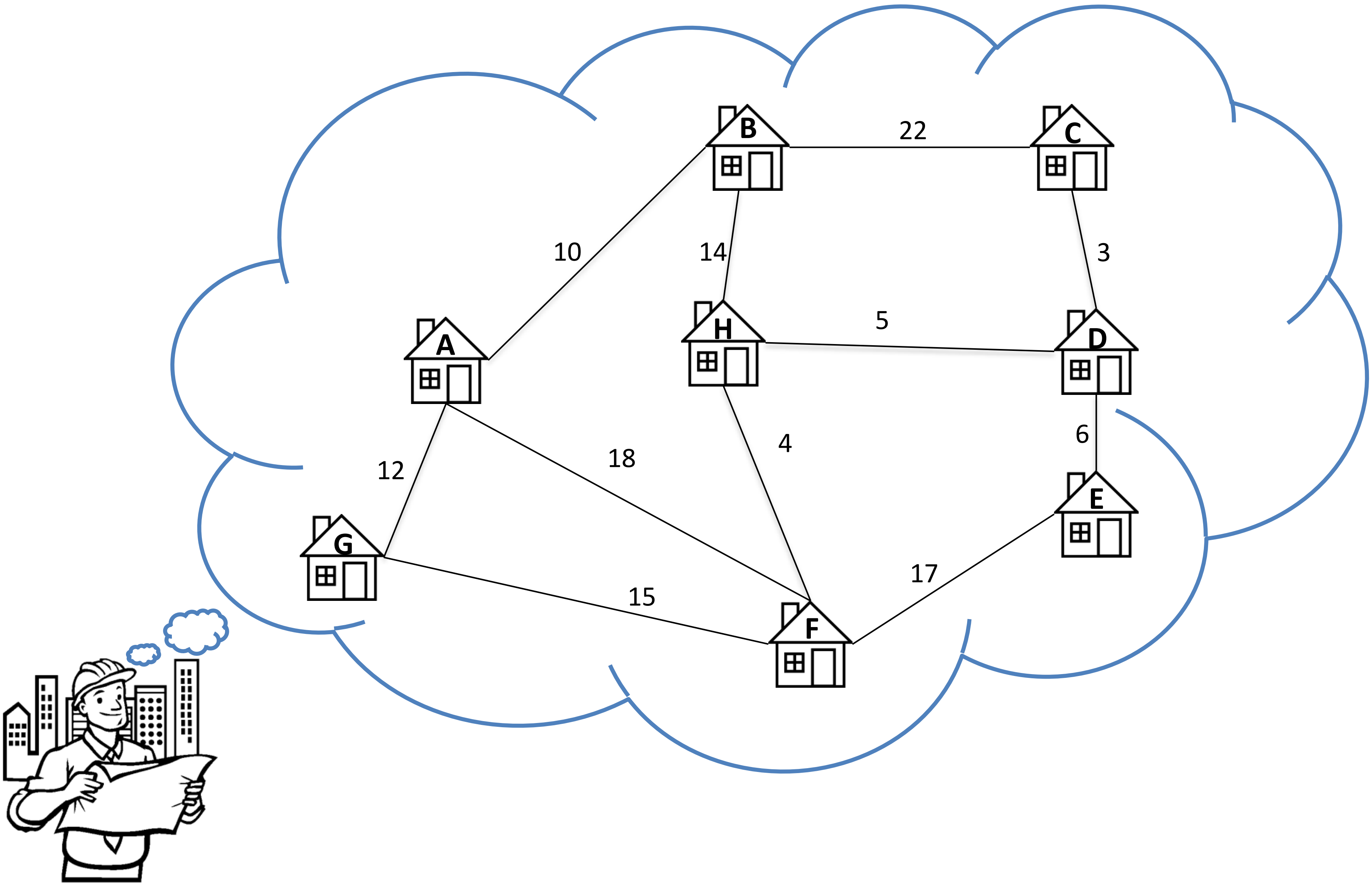
Prim's Algorithm

1. Choose any vertex in the graph
2. Add it to an empty tree
3. Until all nodes are in the tree
   - Choose the edge of least cost that emanates from a node in the tree thus far
   - Add that edge and vertex to the tree

Prim's Algorithm

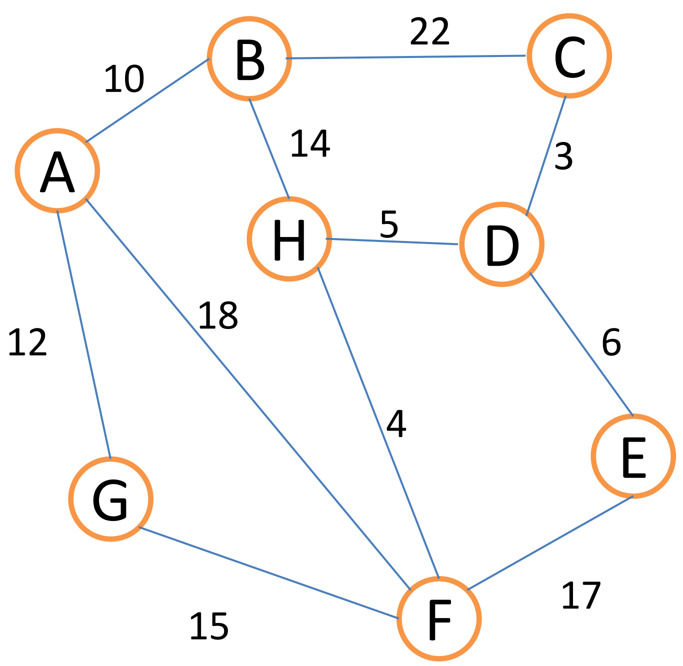
```
Initialize X = {s}; //s∈V is chosen arbitrarily
T = ∅; //Loop invariant: X = vertices spanned by tree-so-
   //far T
while (X ≠ V){
   Let e = (u,v) be the cheapest edge of G with u∈X, v∉X;
   Add e to T;
   Add v to X;
}
//T contains all edges selected in the final minimum
//spanning tree
```

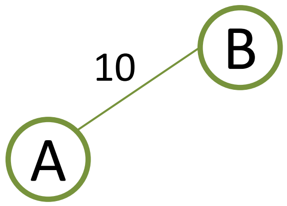
Prim’s Algorithm



Example

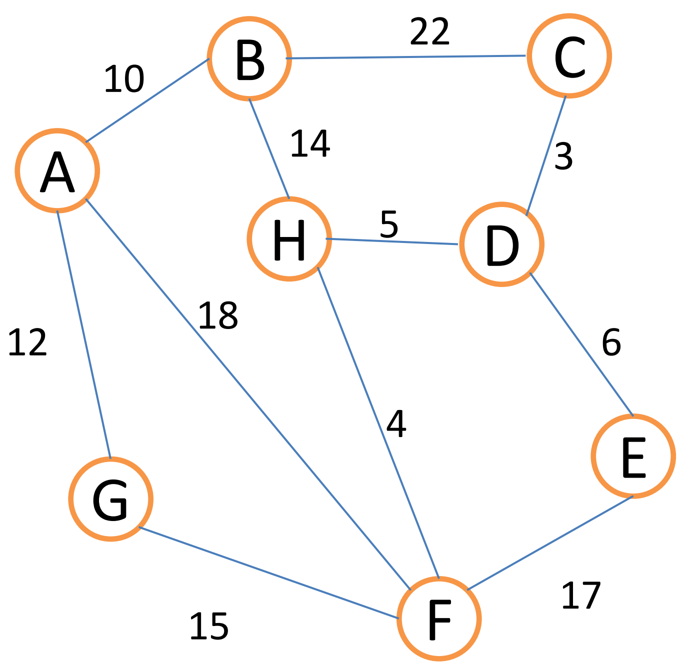
- Find the MST, start with A

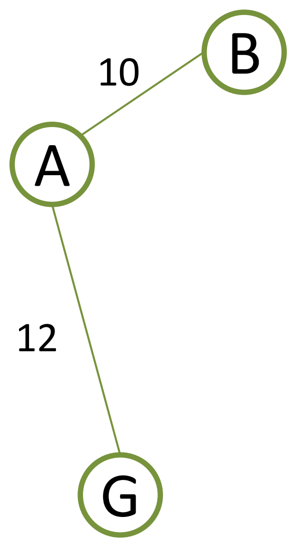




Example

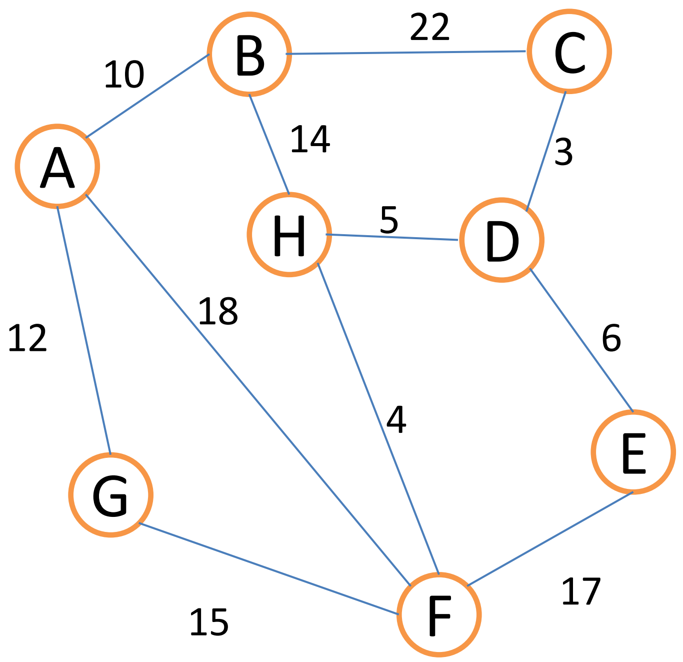
- Find the MST, start with A

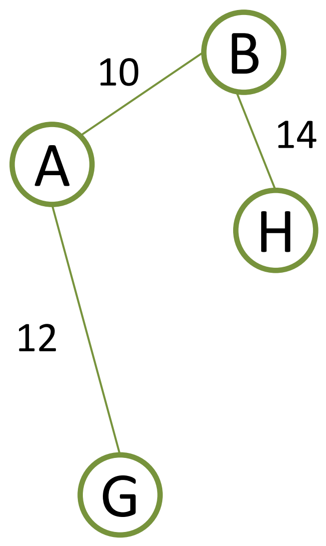




Example

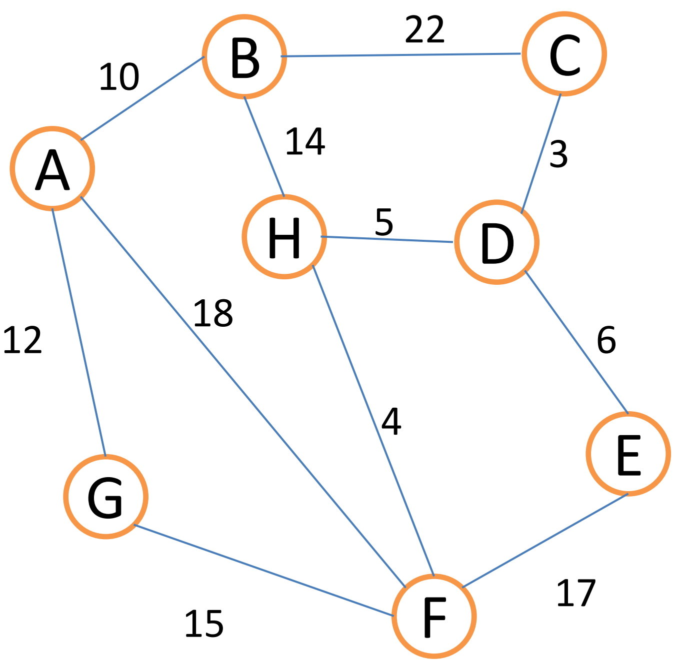
- Find the MST, start with A

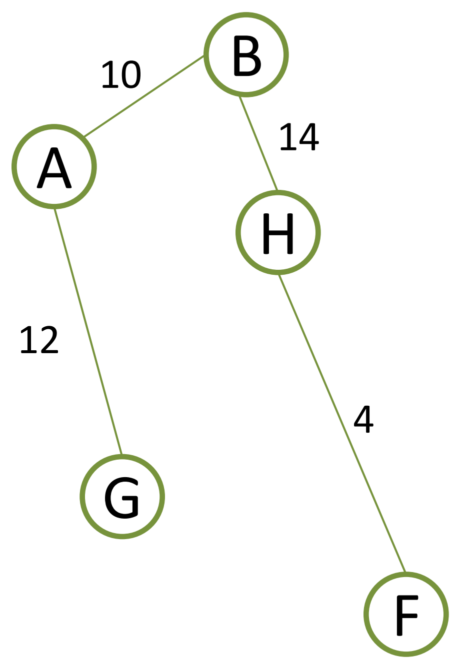




Example

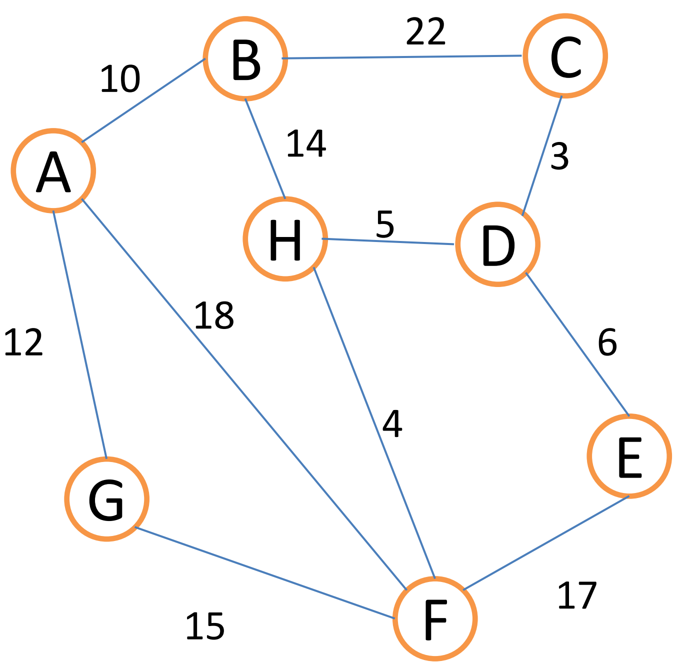
- Find the MST, start with A

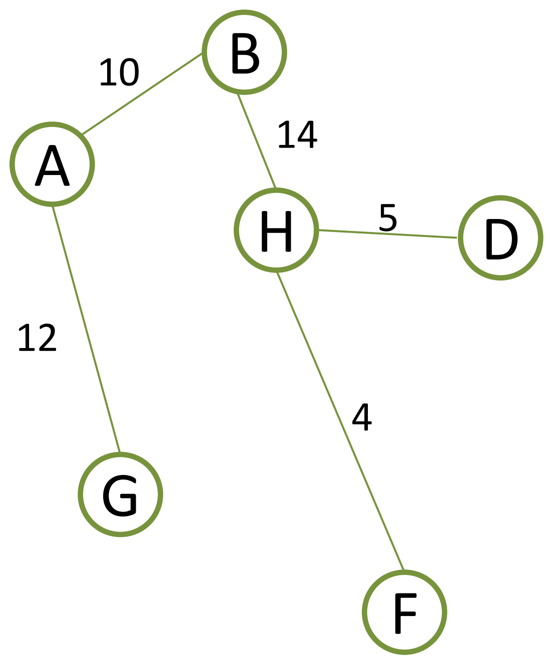




Example

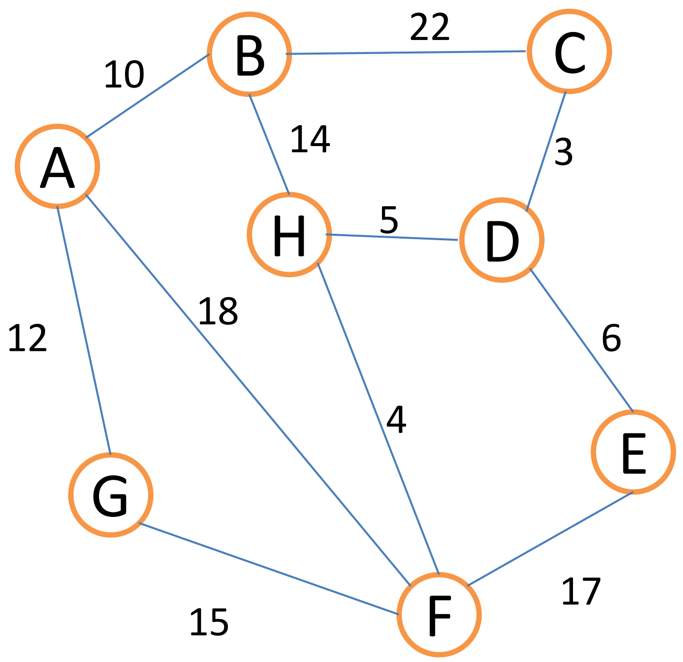
- Find the MST, start with A

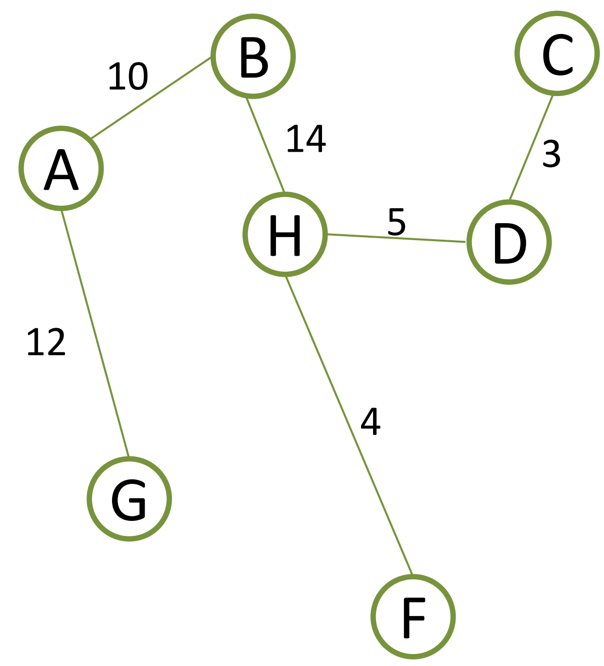




Example

- Find the MST, start with A

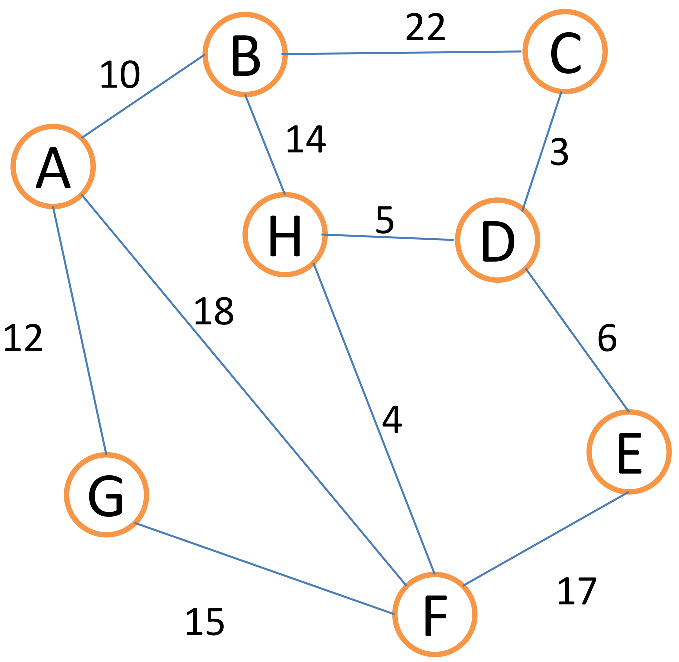


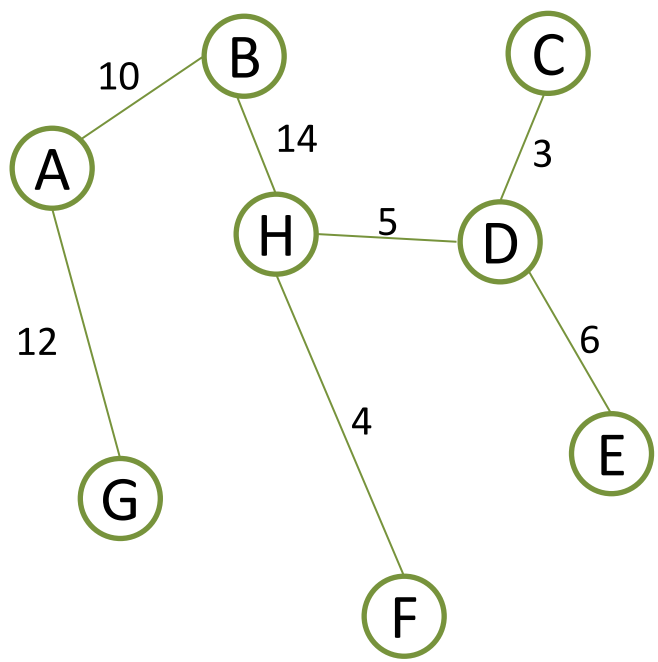


Example

- Find the MST, start with A

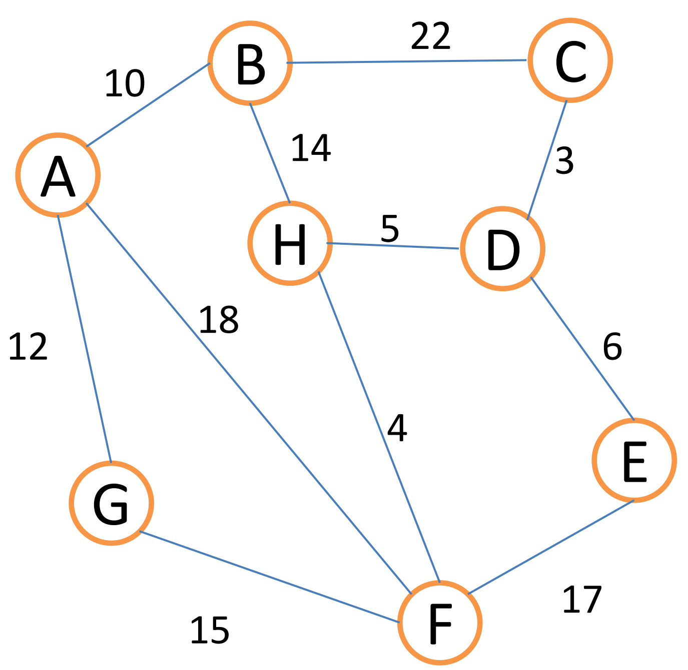
COST:54





Exercise

- Find the MST, start with H



Kruskal’s Algorithm

Kruskal's Algorithm

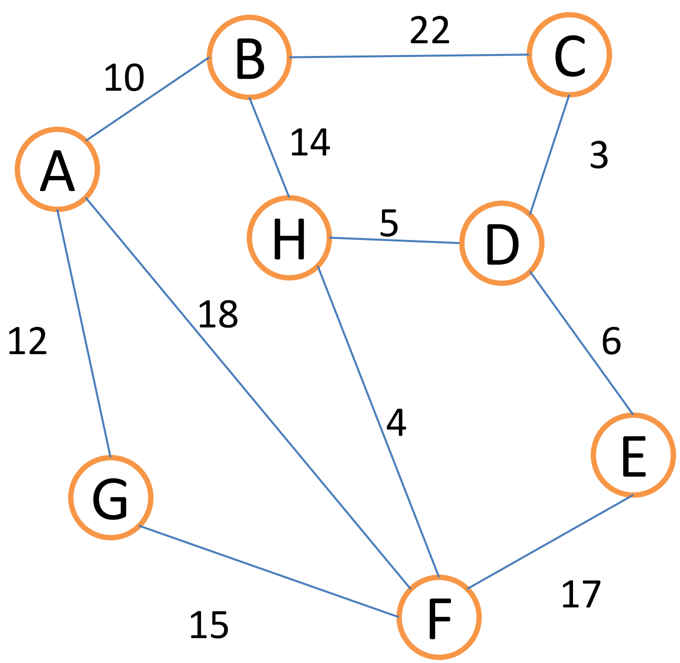
- Forest: Undirected graph, all of whose connected components are trees.
  - Note: A special case of a forest is an empty graph (all connected components are trees with one node)

- Pseudo-Code:

1. Construct a forest from the N nodes in the graph
2. Put the (sorted) edges in a queue
3. Until there are N - 1 edges in the forest (a single tree)
   1. Extract the "cheapest" edge from the queue
   2. If it will form a cycle, discard it
   3. Otherwise, add to the forest (always joins two trees)

Example

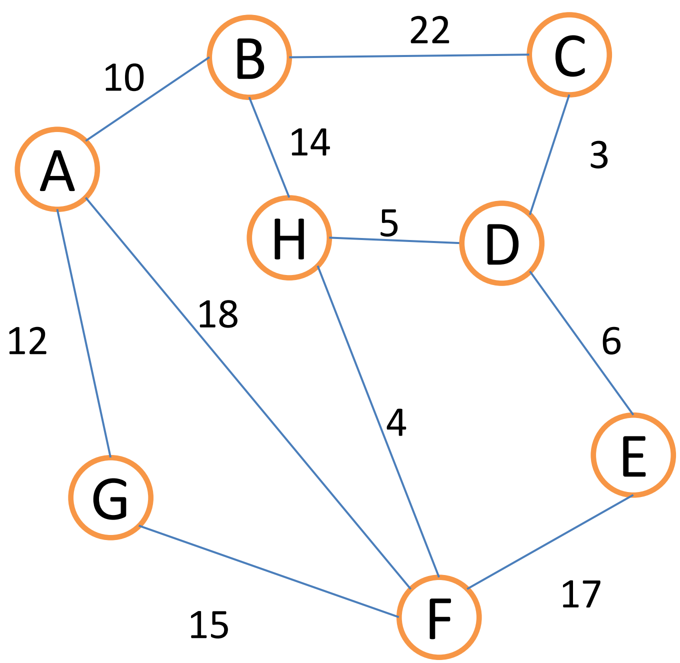
- Compute MST using Kruskal's algorithm

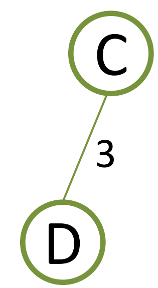


PQ={(C,D),(H,F),(H,D),(D,E),(A,B),(A,G),(B,H),(G,F),(F,E),(A,F),(B,C)}

Example

- Compute MST using Kruskal's algorithm

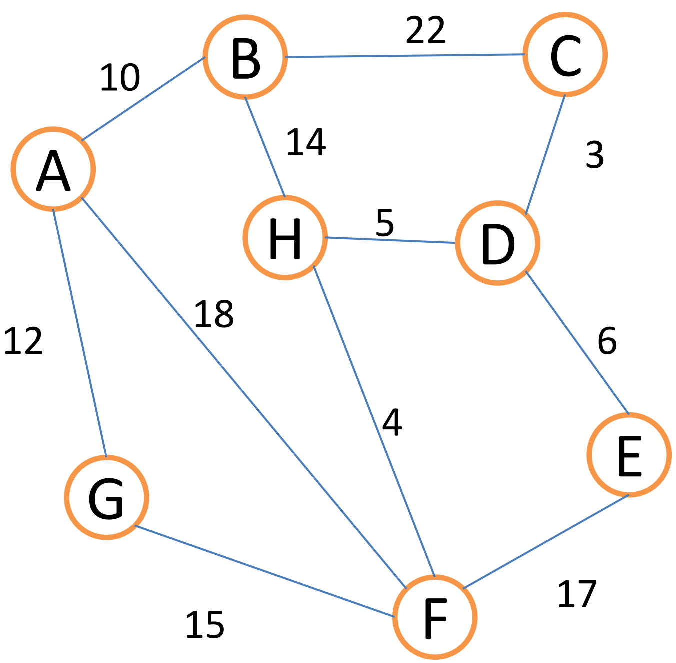


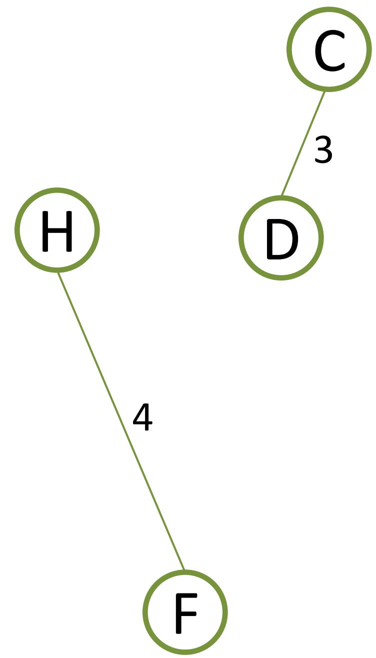


PQ={ ~~(C,D)~~, (H,F), (H,D), (D,E), (A,B), (A,G), (B,H), (G,F), (F,E), (A,F), (B,C) }

Example

- Compute MST using Kruskal's algorithm

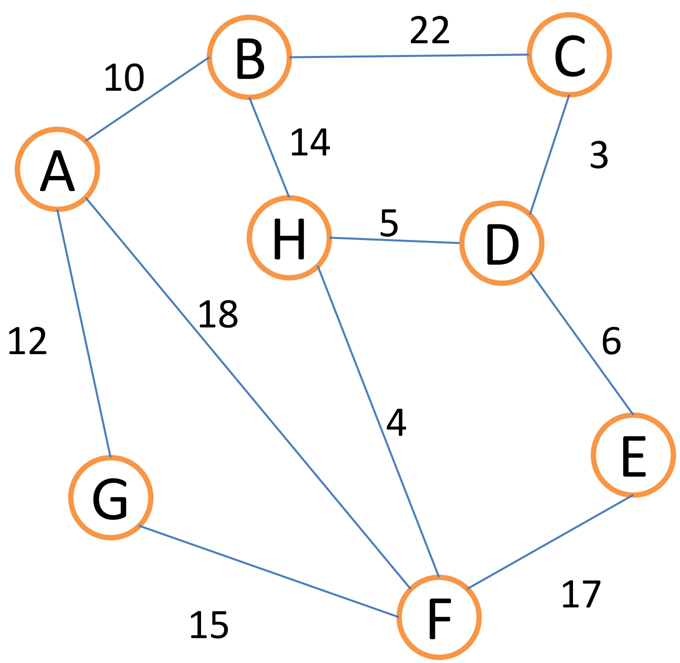


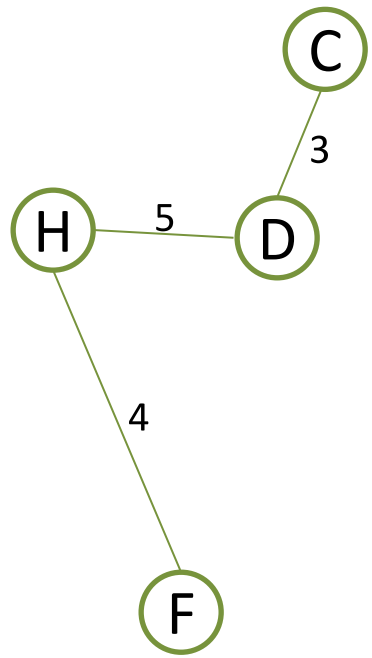


PQ={ ~~(C,D)~~, ~~(H,F)~~, (H,D), (D,E), (A,B), (A,G), (B,H), (G,F), (F,E), (A,F), (B,C) }

Example

- Compute MST using Kruskal's algorithm

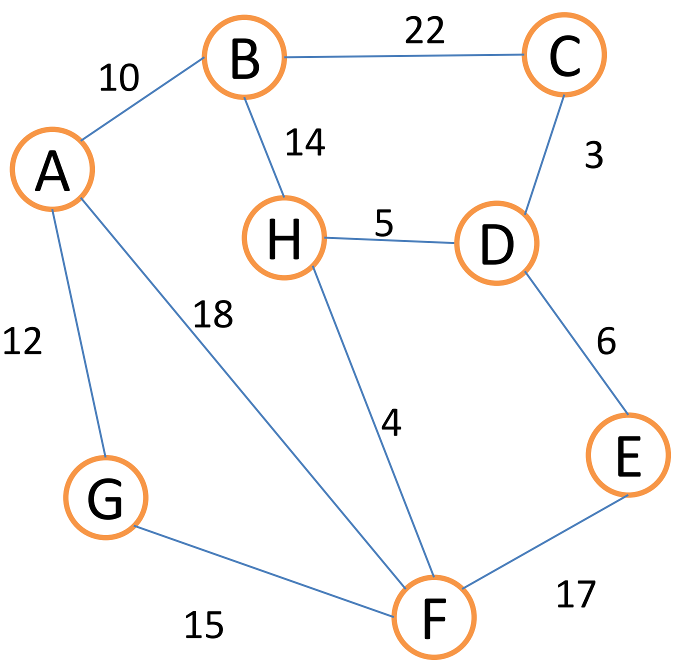


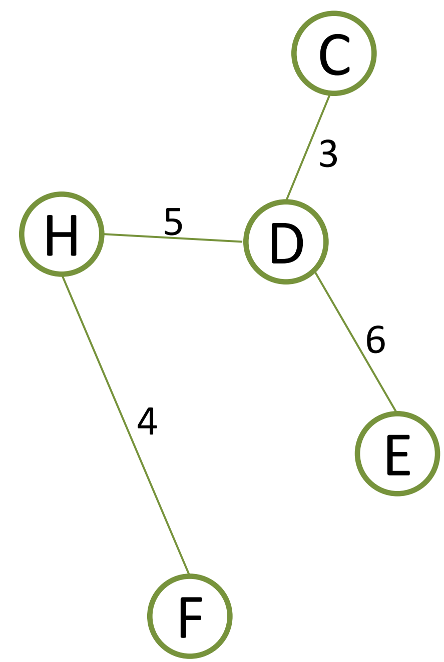


PQ={ ~~(C,D)~~, ~~(H,F)~~, ~~(H,D)~~, (D,E), (A,B), (A,G), (B,H), (G,F), (F,E), (A,F), (B,C) }

Example

- Compute MST using Kruskal's algorithm

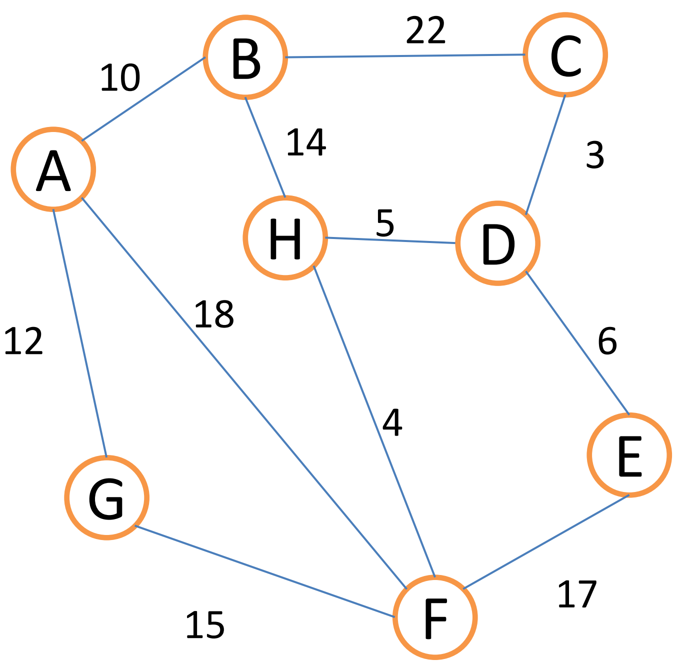


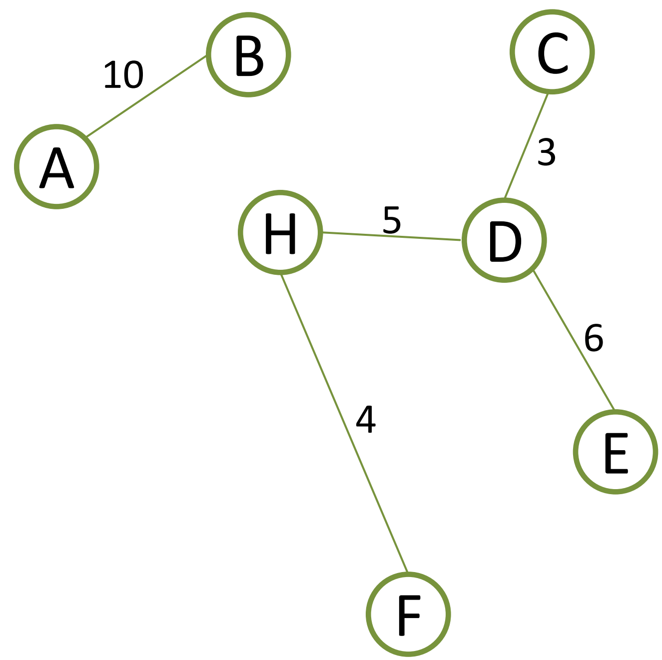


PQ={ ~~(C,D)~~, ~~(H,F)~~, ~~(H,D)~~, ~~(D,E)~~, (A,B), (A,G), (B,H), (G,F), (F,E), (A,F), (B,C) }

Example

- Compute MST using Kruskal's algorithm

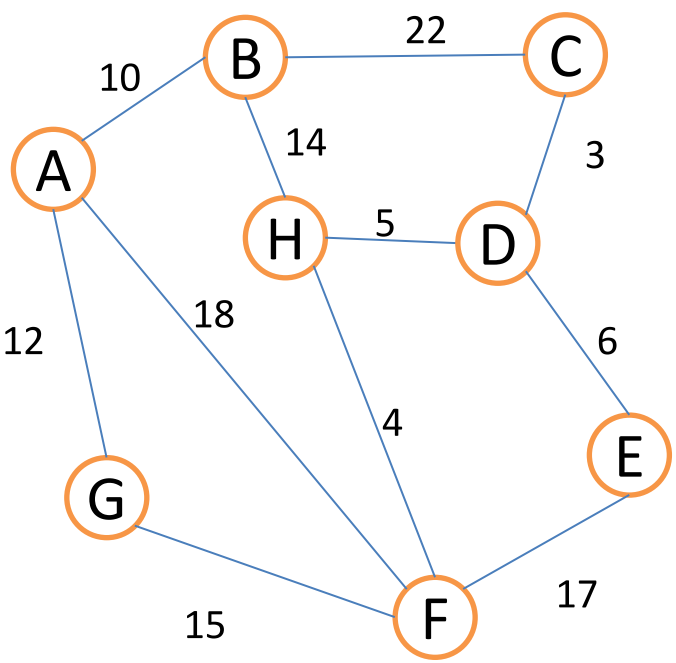


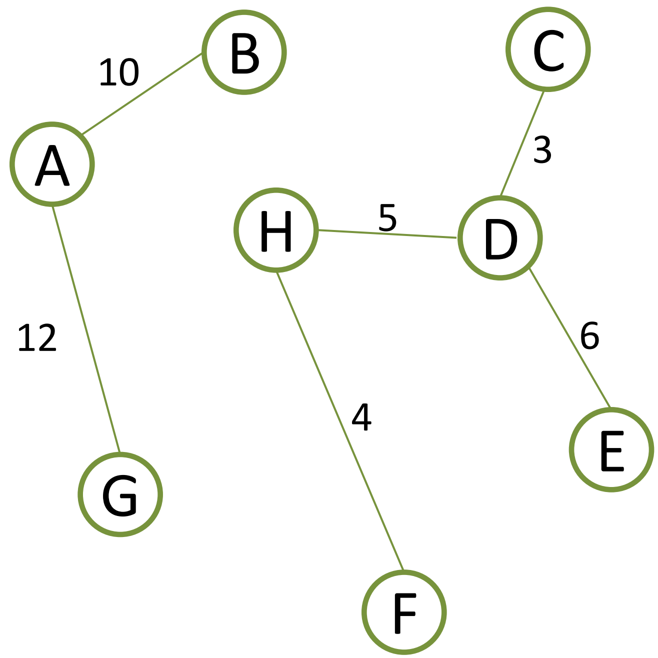


PQ={ ~~(C,D)~~, ~~(H,F)~~, ~~(H,D)~~, ~~(D,E)~~, ~~(A,B)~~, (A,G), (B,H), (G,F), (F,E), (A,F), (B,C) }

Example

- Compute MST using Kruskal's algorithm

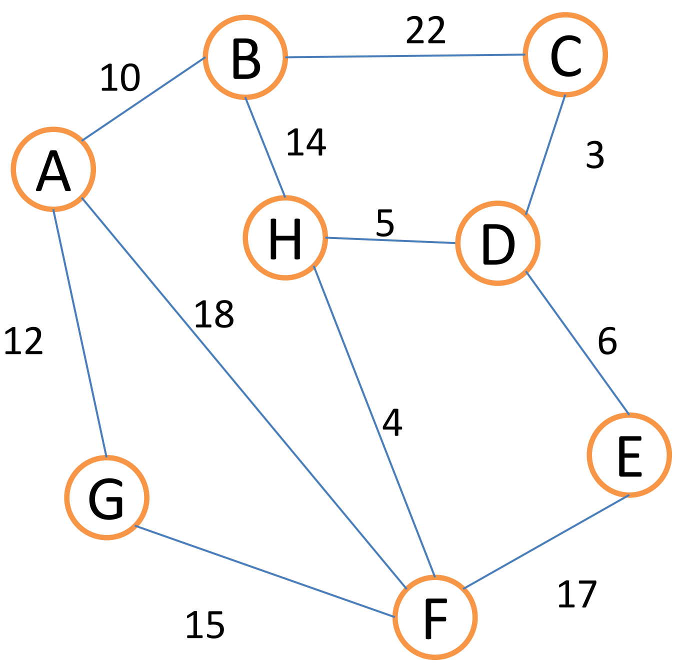


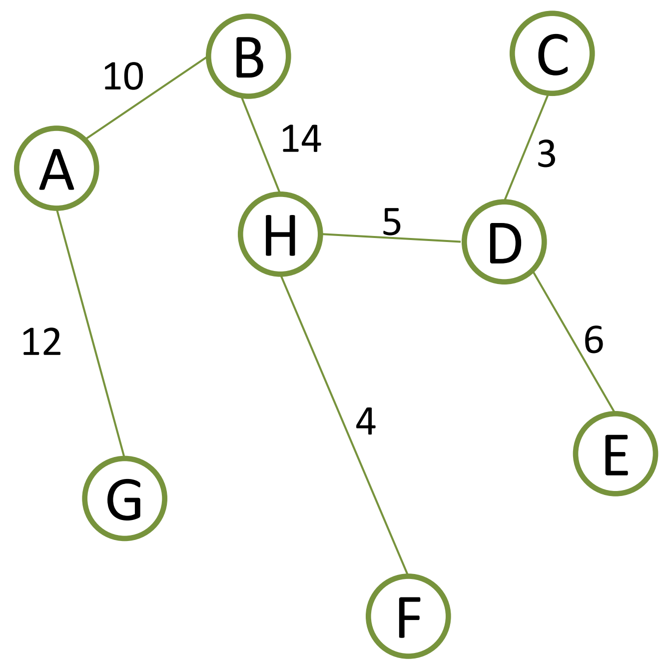


PQ={ ~~(C,D)~~, ~~(H,F)~~, ~~(H,D)~~, ~~(D,E)~~, ~~(A,B)~~, ~~(A,G)~~, (B,H), (G,F), (F,E), (A,F), (B,C) }

Example

- Compute MST using Kruskal’s algorithm

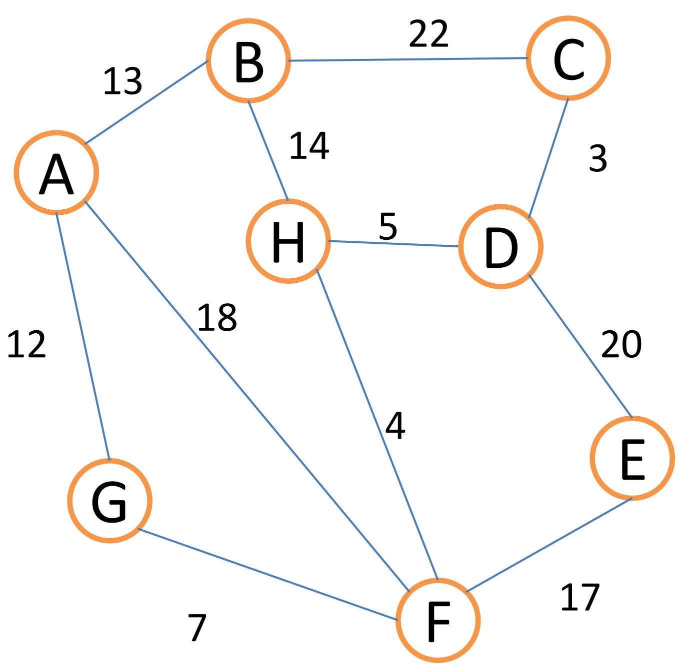




PQ={~~(C,D)~~,~~(H,F)~~,~~(H,D)~~,~~(D,E)~~,~~(A,B)~~,~~(A,G)~~,~~(B,H)~~,(G,F),(F,E),(A,F),(B,C)}

Exercise

- Compute MST using Kruskal's algorithm



Kruskal’s Algorithm

- Complexity?
- |E| = |V -1|
- O(ExtractMin)
  - Arrays: O(E)
  - Heap: log(E)
- O(V.ExtractMin)
  - Arrays:O(V^2)
  - Heap: O(VlogV)

Considerations

- The efficiency of the algorithms depends on the implementation of the “auxiliary” data structures as well as the density of the graphs.

- If all the edges from a node have unique weights, the resulting tree will be unique. (Otherwise, there could be multiple min/max spanning trees.)

- Both algorithms are greedy algorithms.

Summary

- Spanning tree
  - Minimum spanning tree
  - Prim’s algorithm
  - Kruskal’s algorithm