

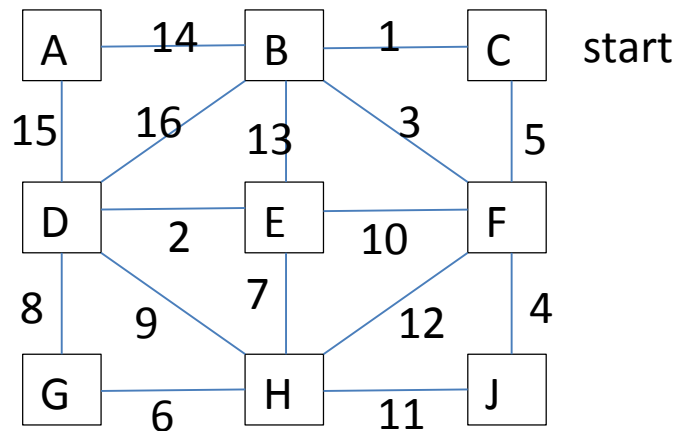
SSSP

Single-Source Shortest Paths:

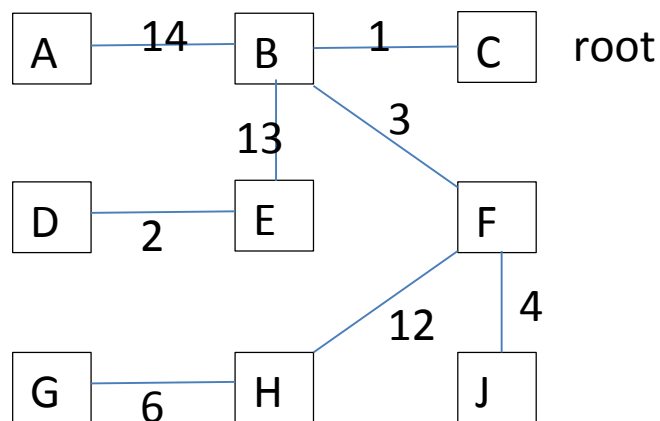
Given a weighted (undirected or directed) graph and a specified start vertex S , find a minimum-distance path from S to each destination vertex

[Restriction: every edge weight $w(x,y) \geq 0$]

Example, with start vertex = C:



Shortest
paths
tree:



Dijkstra's algorithm for finding a shortest paths tree in (undirected or directed) graph G:

choose a start vertex;

H = new MinHeap();

for each vertex v {

 if (v==start) cost[v] = 0;

 else cost[v] = ∞ ;

 H.insert (v, cost[v]); // cost[v] is the key

}

while (! H.isEmpty()) {

 x = H.removeMin();

 for each vertex y such that (x,y) is an edge in graph G

 if (cost[x] + weight(x,y) < cost[y]) {

 cost[y] = cost[x] + weight(x,y);

 parent[y] = x;

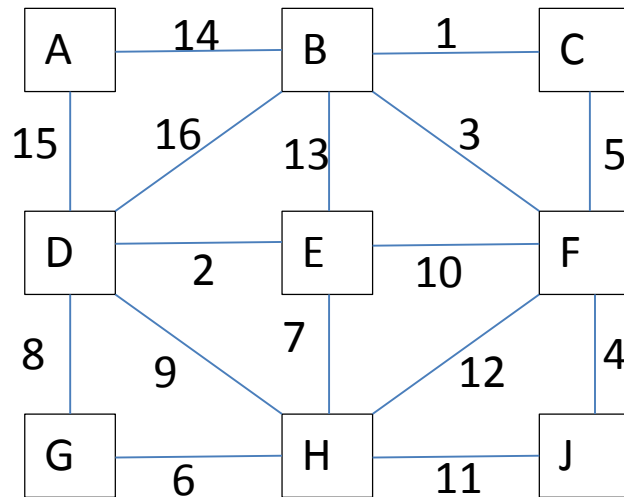
 H.decreaseKey (y, cost[y]);

 // swap y up the heap as necessary

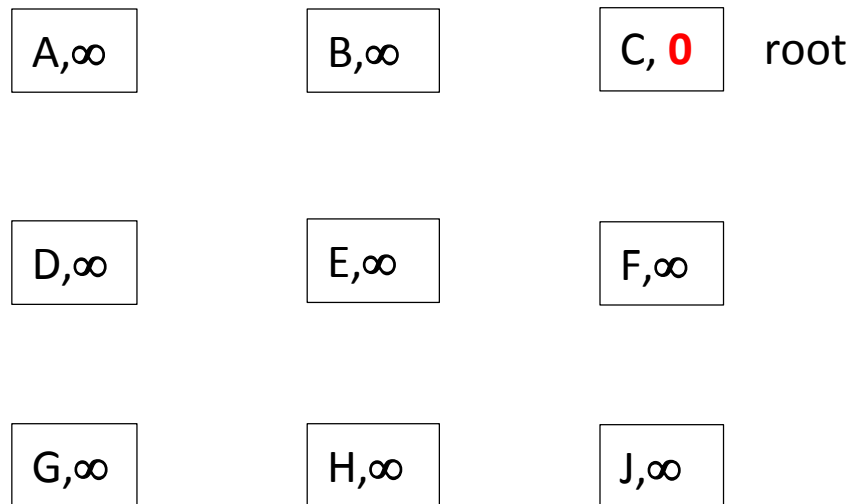
 }

}

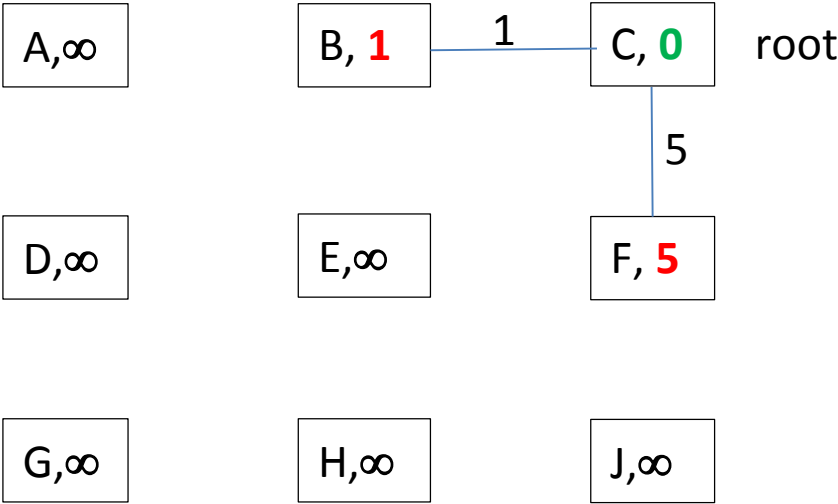
Trace Dijkstra's algorithm:



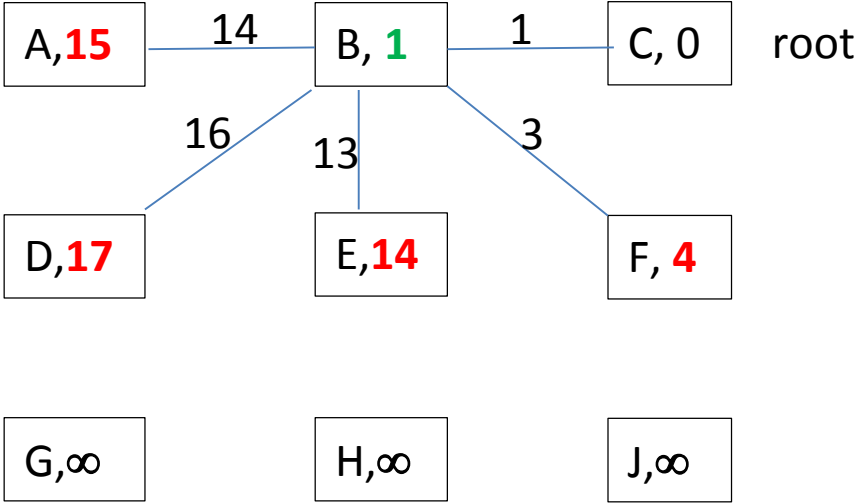
Let start vertex = C



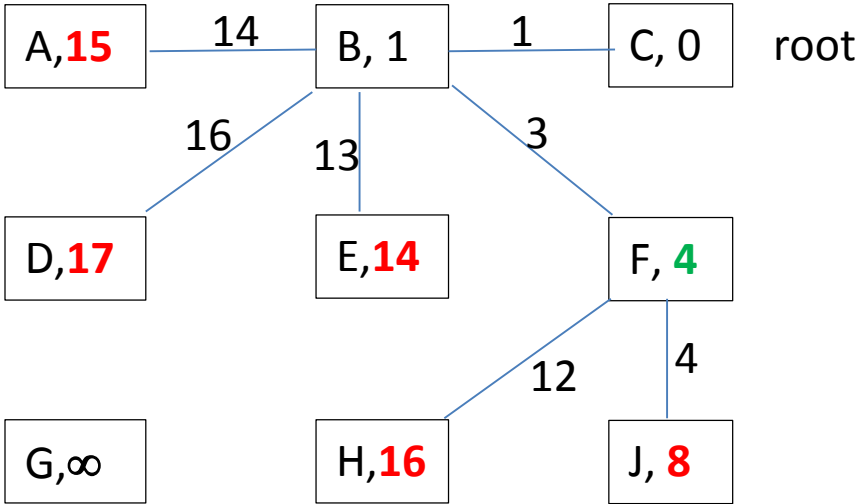
RemoveMin: C



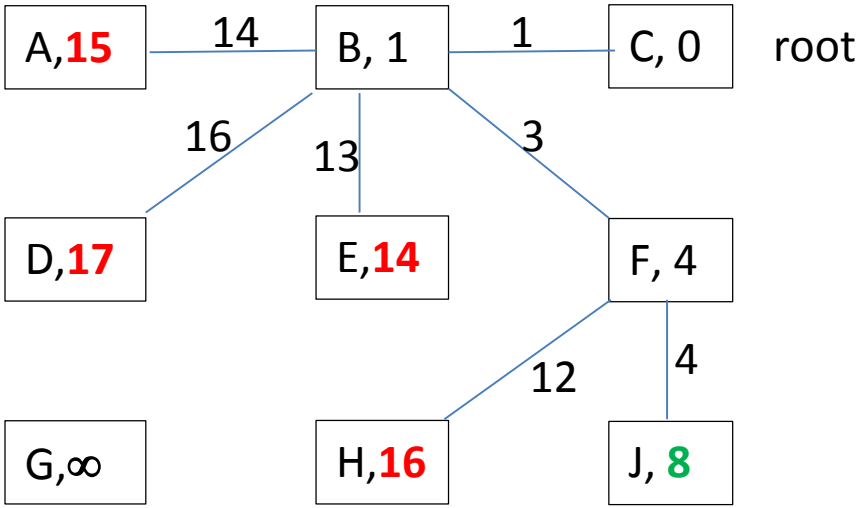
RemoveMin: B



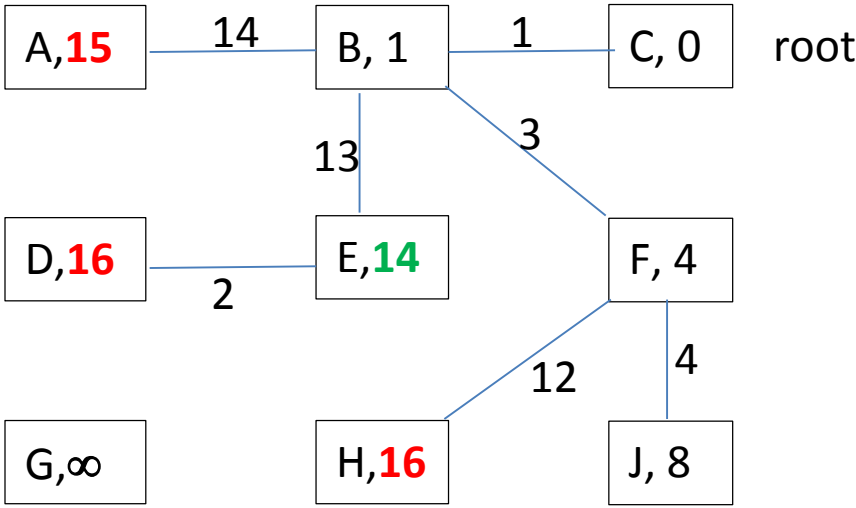
RemoveMin: F



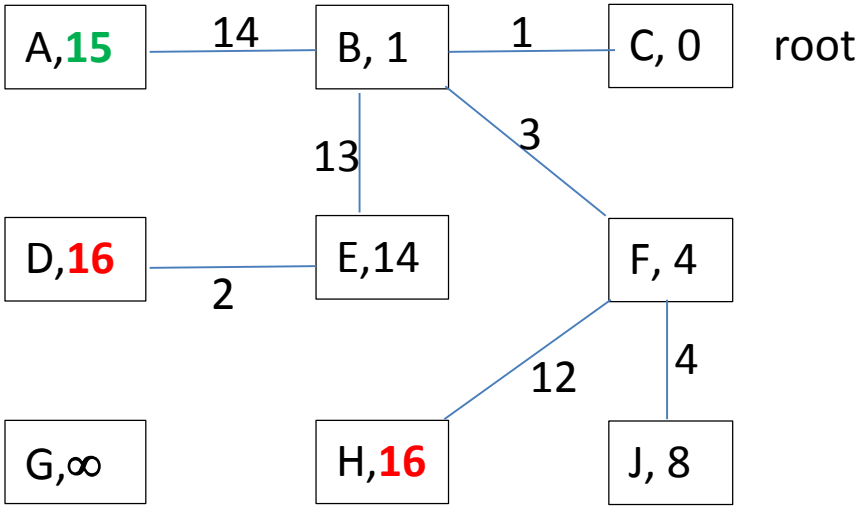
RemoveMin: J

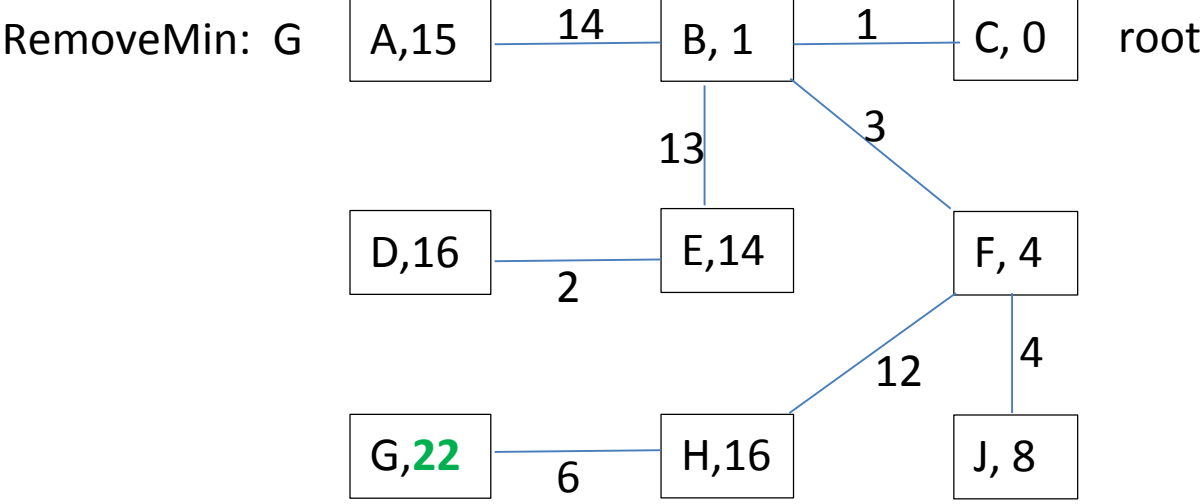
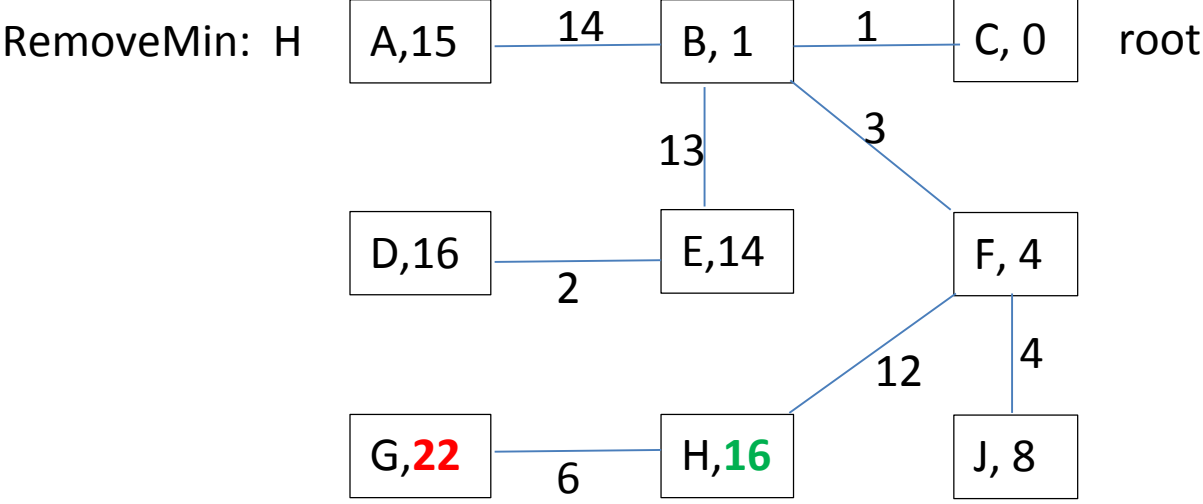
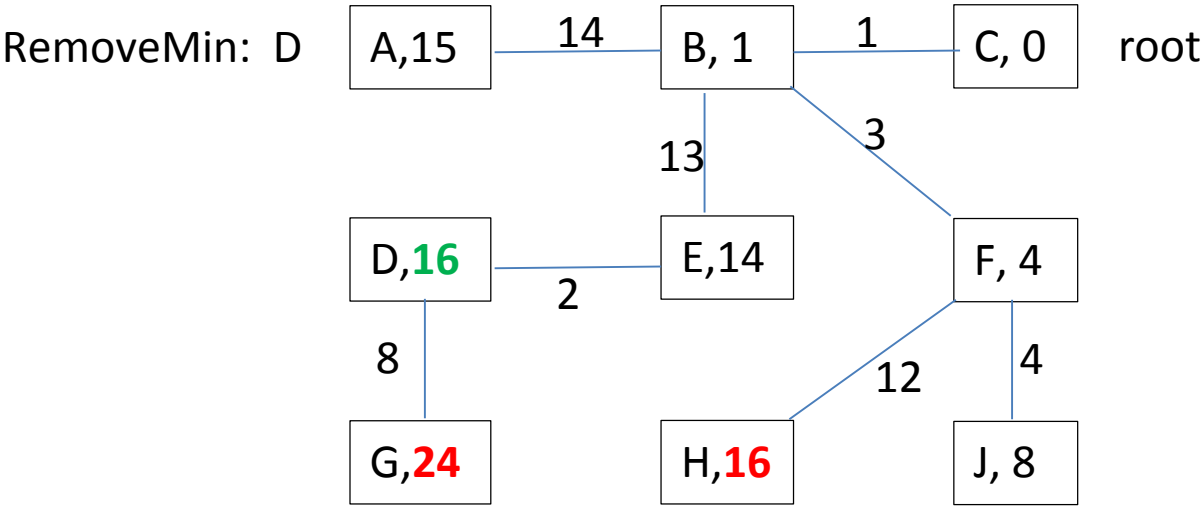


RemoveMin: E



RemoveMin: A





Implementation and analysis of Dijkstra's algorithm:

- (i) Use adjacency lists representation for graph G
- Time to find all edges (x,y) in G is $\theta(n+m)$ time

Use binary heap H

- n inserts, each takes $\theta(\lg n)$ time
- n removeMins, each takes $\theta(\lg n)$ time
- $\leq m$ decreaseKeys, each takes $\theta(\lg n)$ time
- Time for all heap operations is $\theta((2n+m) \lg n)$ time

Total time for Dijkstra using method (i) is $\theta(m \lg n)$

- (ii) Use adjacency matrix representation for graph G
- Time to find all edges (x,y) in G is $\theta(n^2)$ time

Use boolean array in place of heap H:

array[v] = true if v is currently in the heap

- n inserts, each takes $\theta(1)$ time
- n removeMins, each takes $\theta(n)$ time
- $\leq m$ decreaseKeys, each takes $\theta(1)$ time
- Time for all heap operations is $\theta(n^2)$ time

Total time for Dijkstra using method (ii) is $\theta(n^2)$