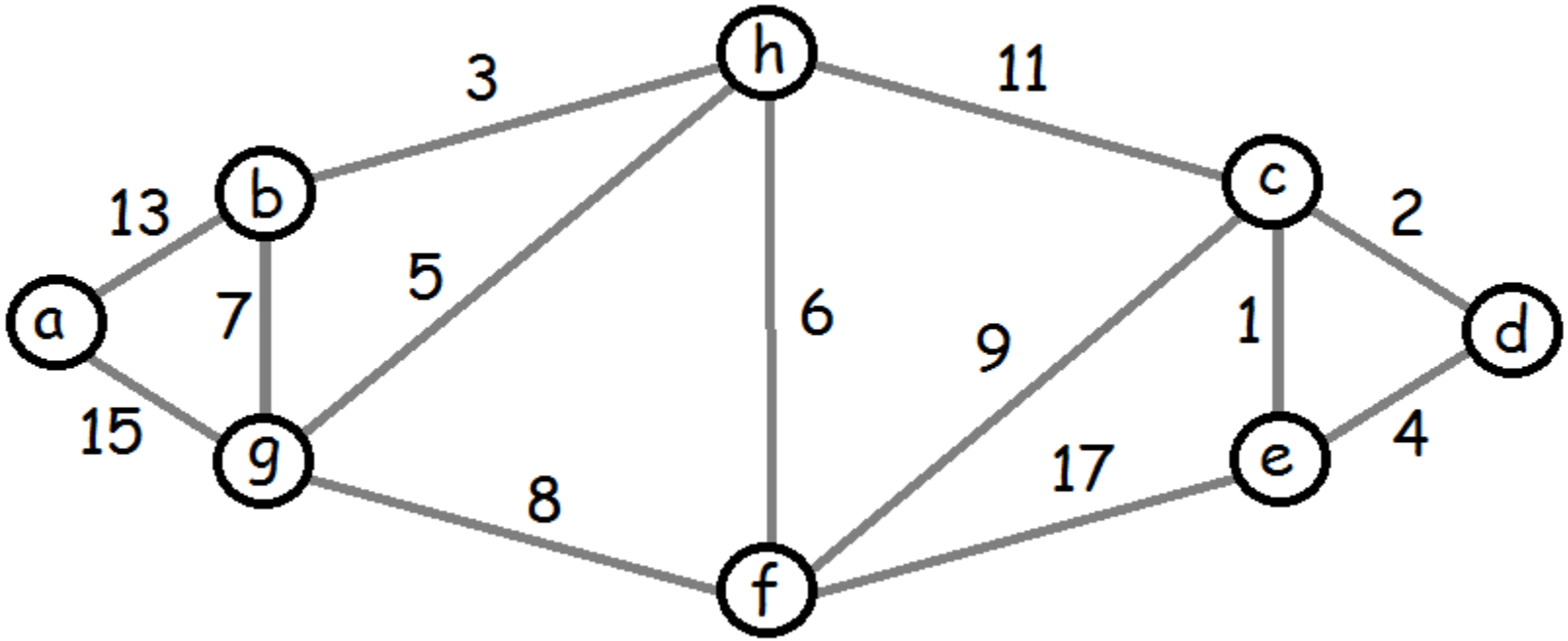
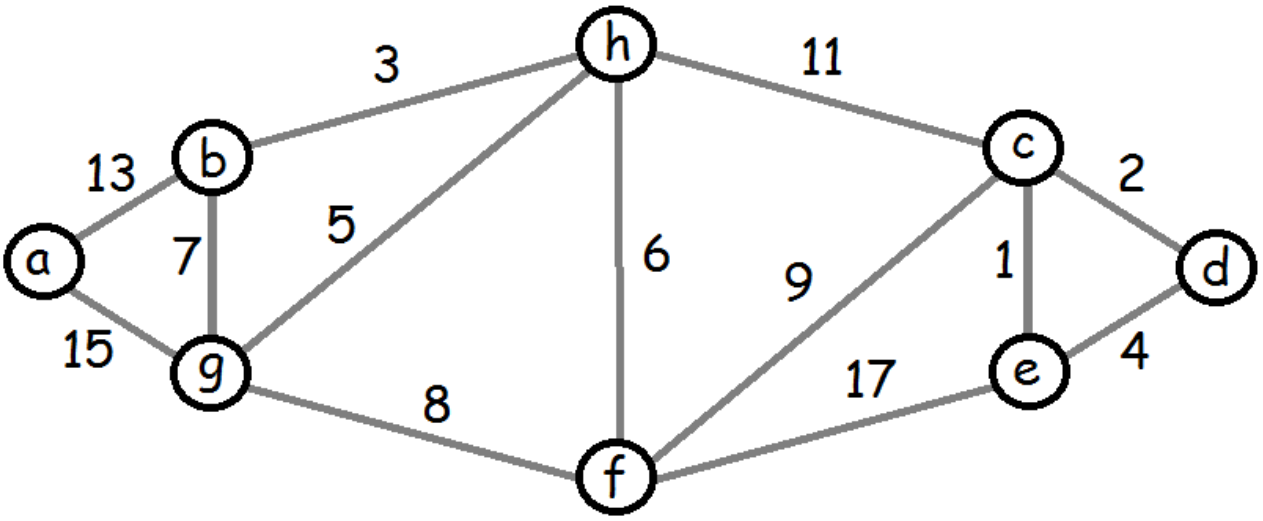
1. For Kruskal's algorithm, number the MST edges with 1, 2, 3, 4, 5, 6, 7 to indicate the order in which they are added into the MST.

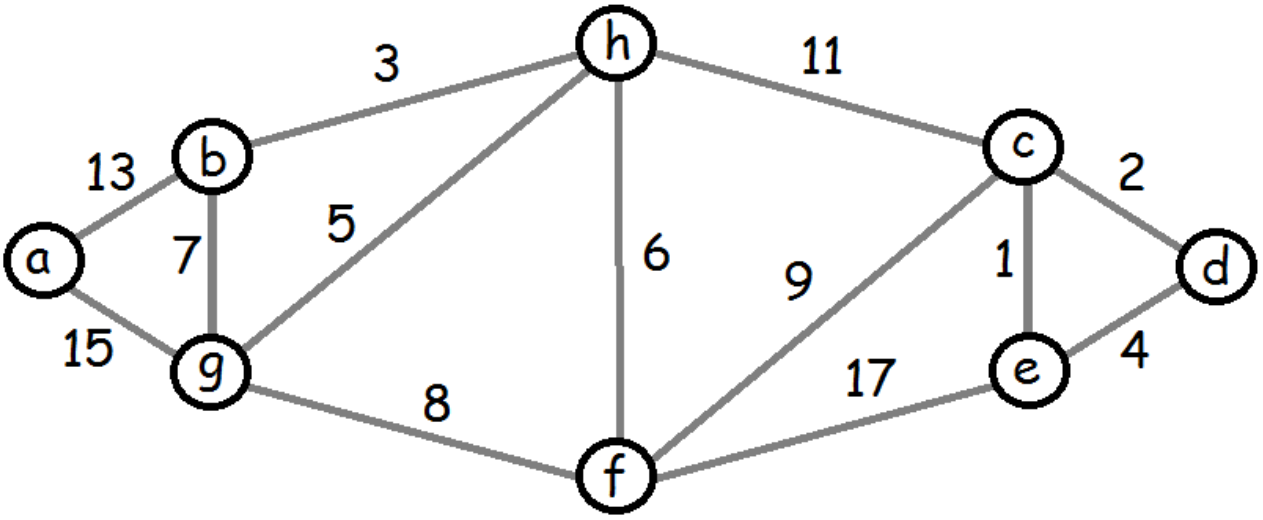


2. For the Dijkstra/Prim algorithm, number the MST edges with 1, 2, 3, 4, 5, 6, 7 to indicate the order in which they are added into the MST. Start at vertex a.

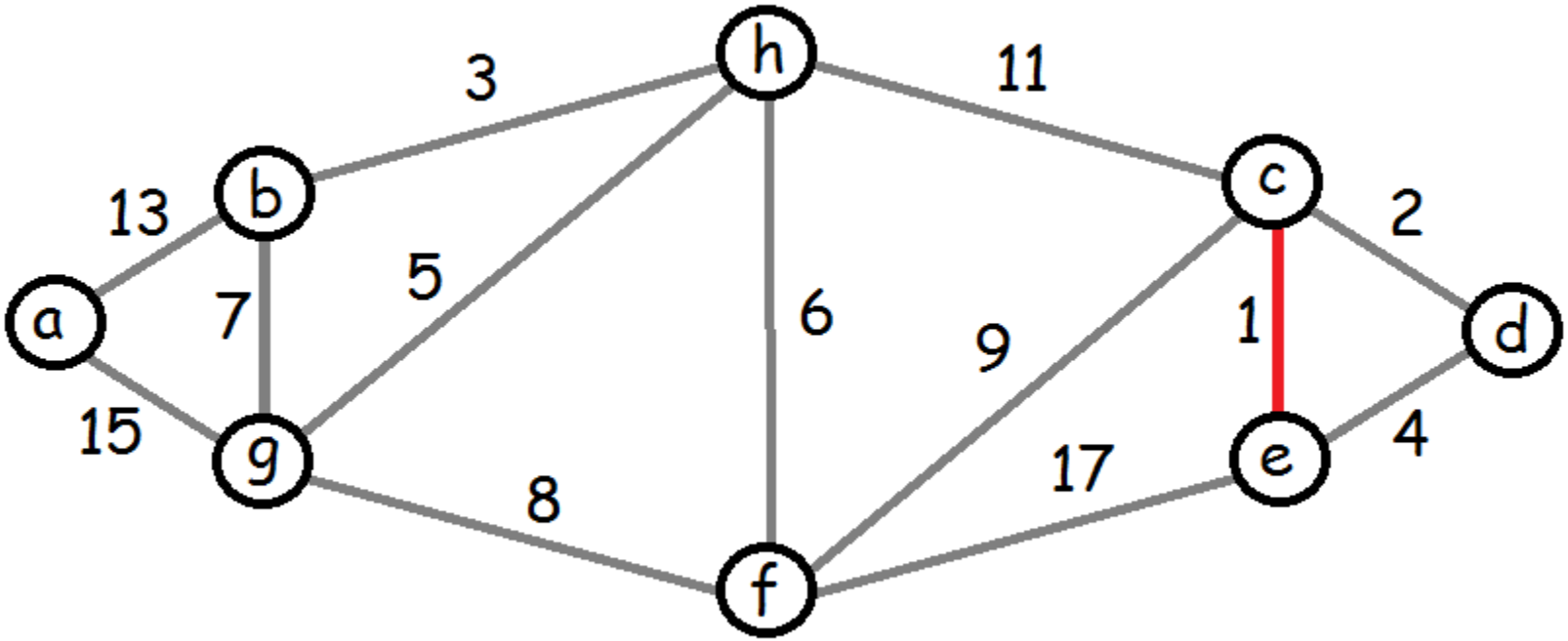
Kruskal's algorithm:



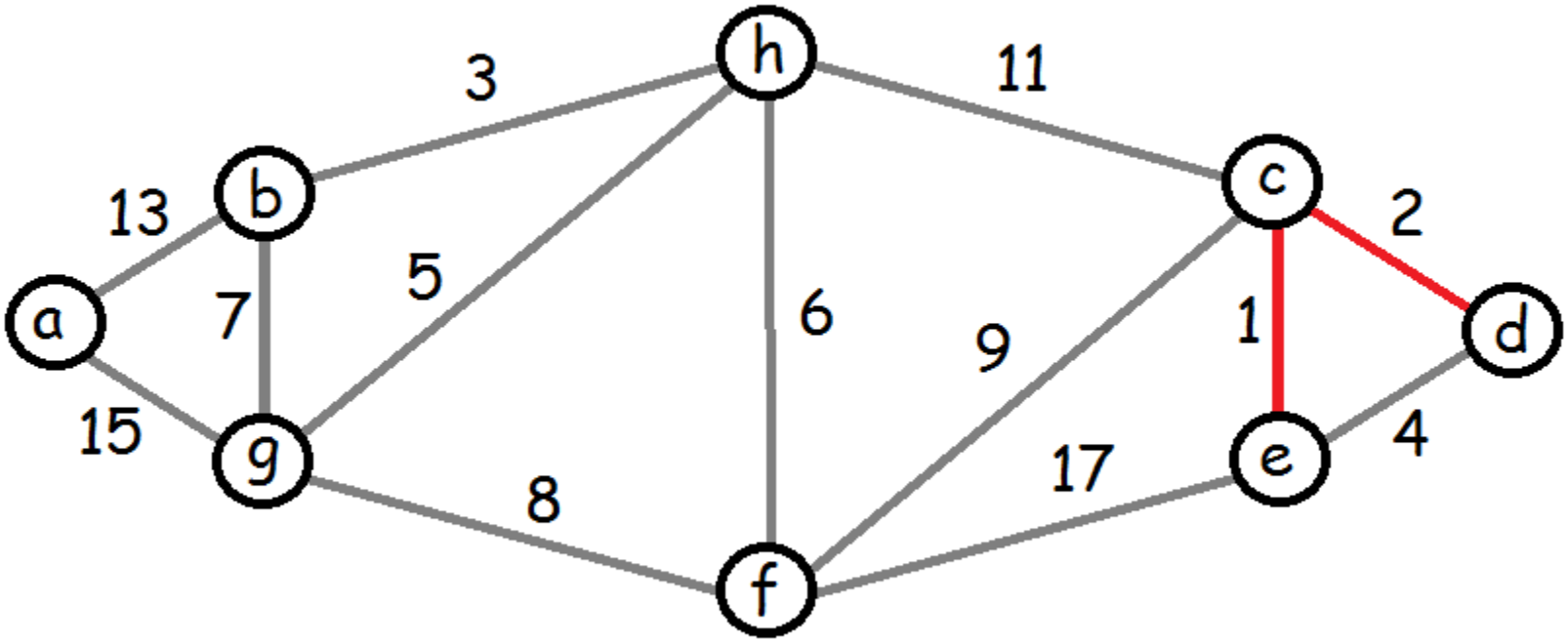
Dijkstra/Prim algorithm:



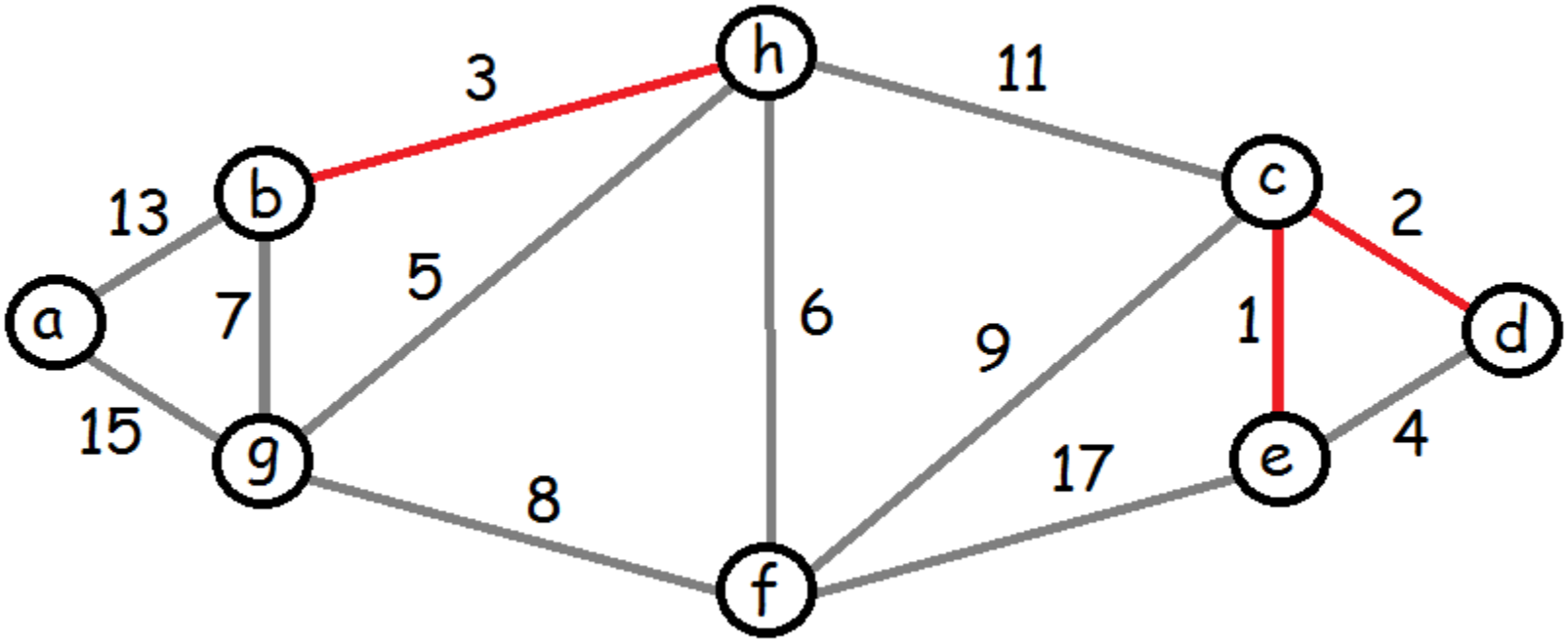
Kruskal's algorithm:



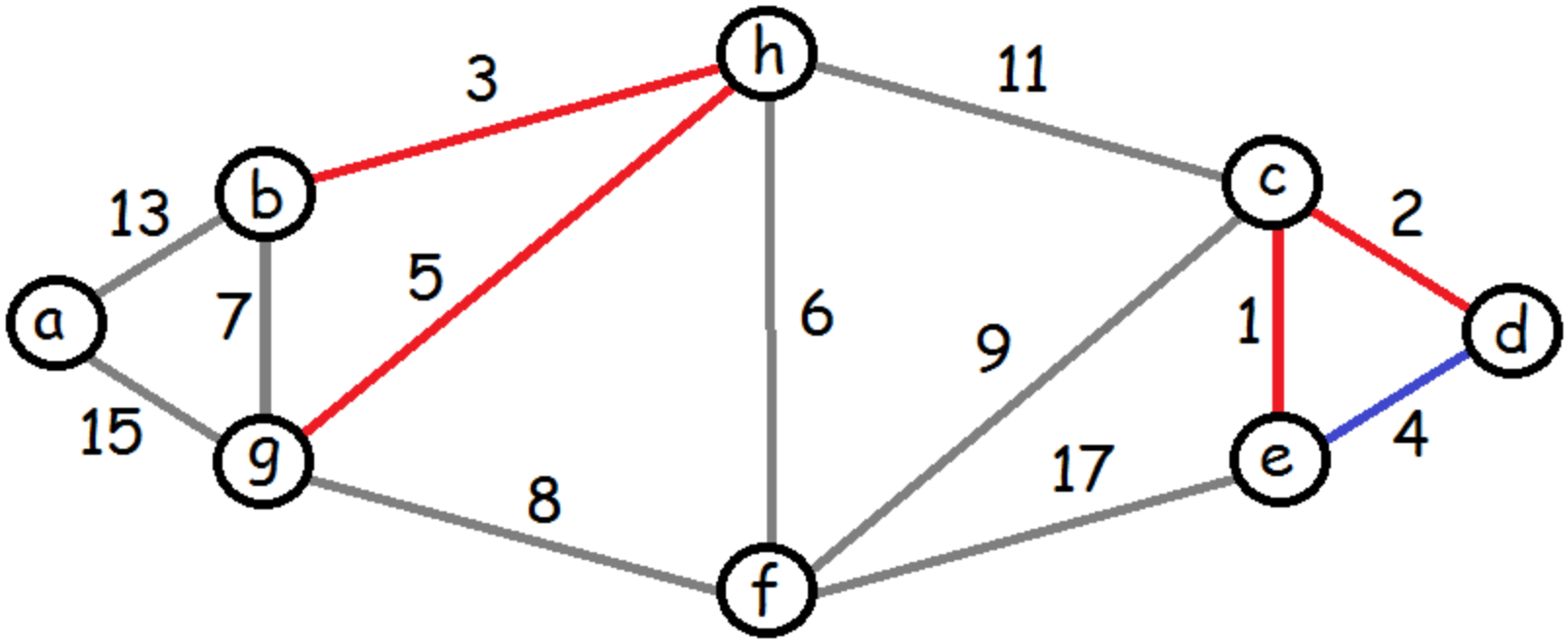
Kruskal's algorithm:



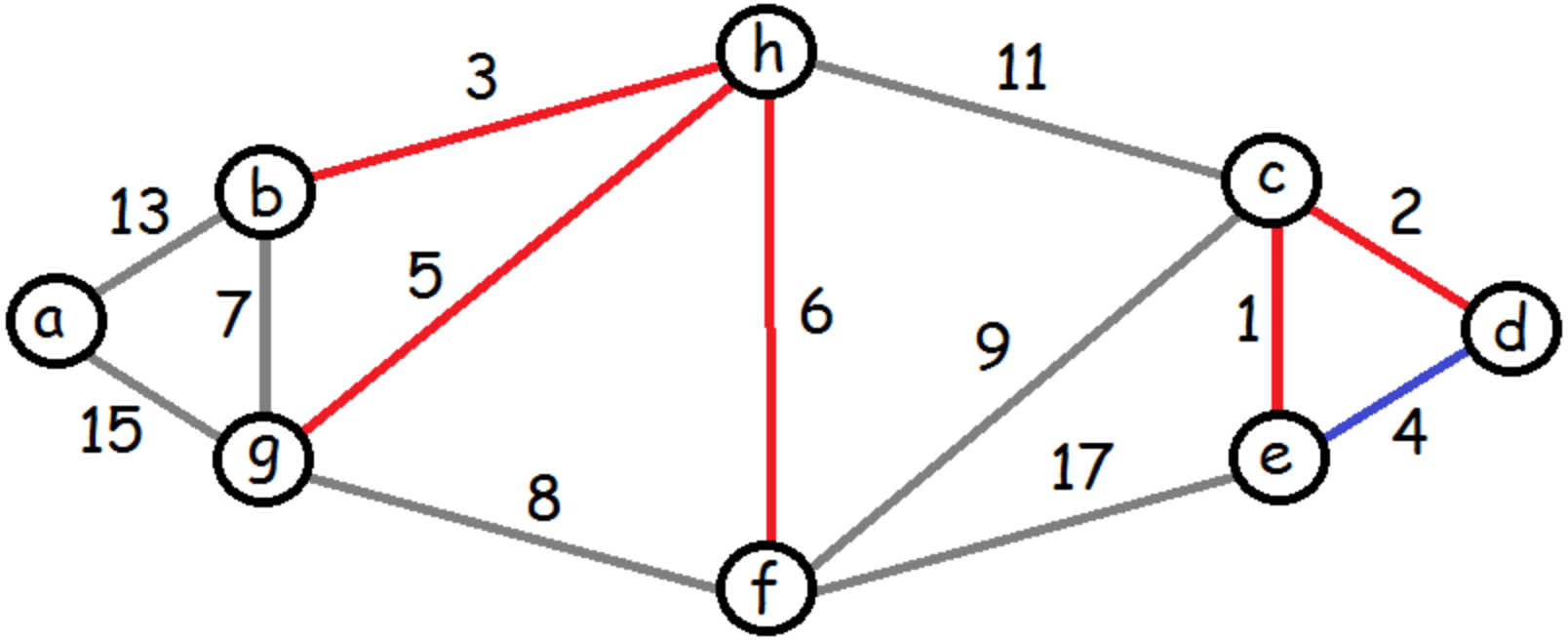
Kruskal's algorithm:



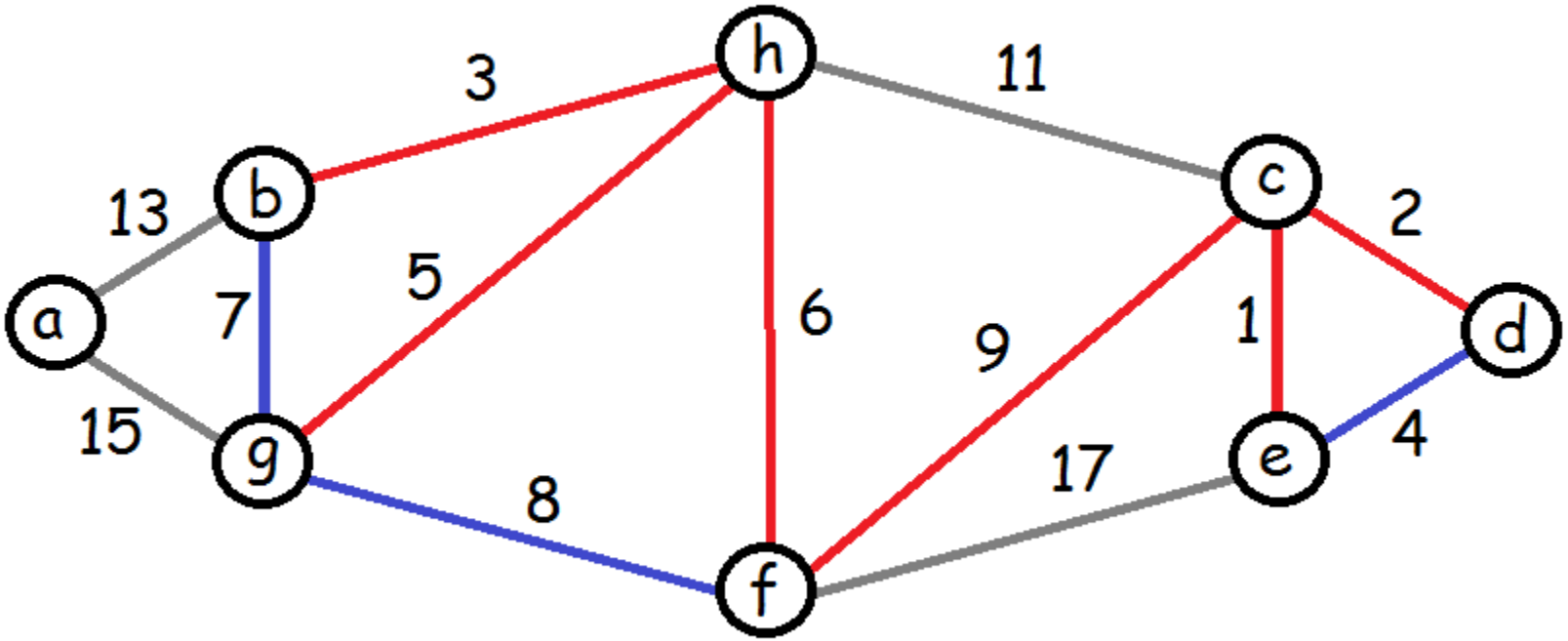
Kruskal's algorithm:



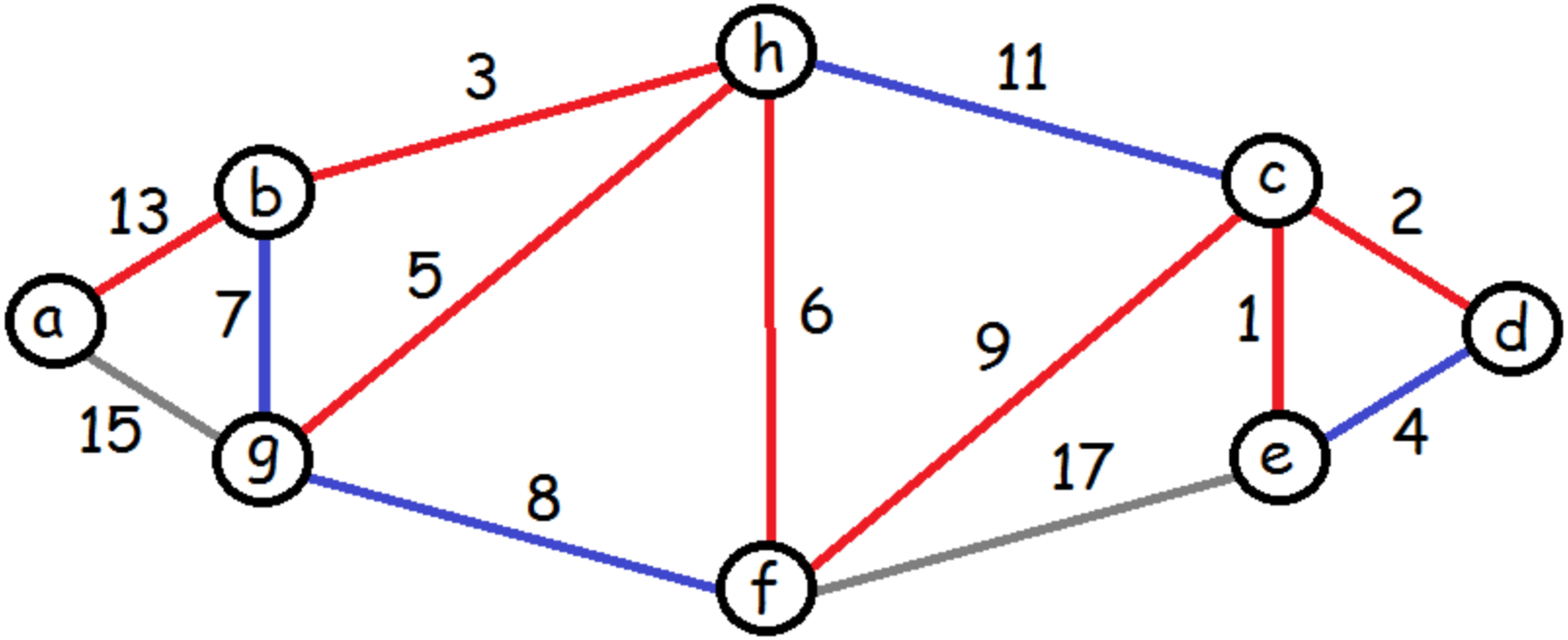
Kruskal's algorithm:



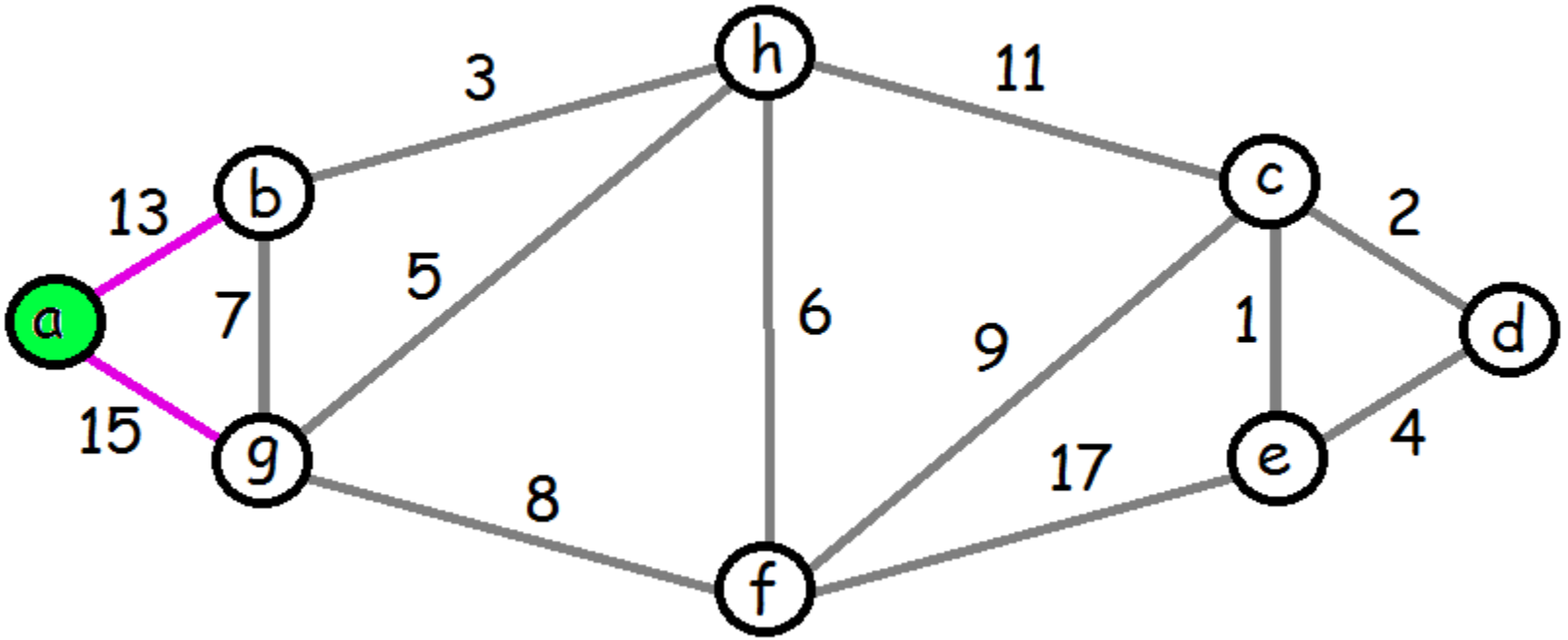
Kruskal's algorithm:



Kruskal's algorithm:



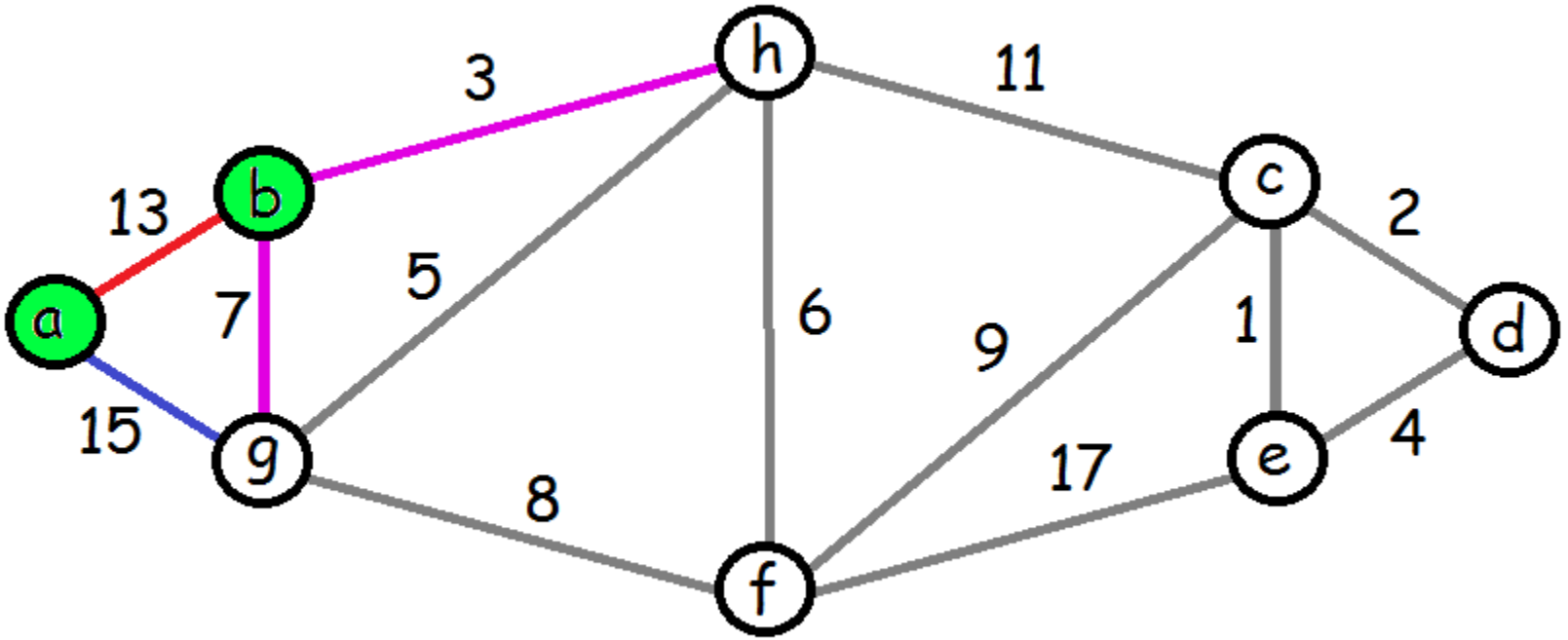
Dijkstra/Prim algorithm:



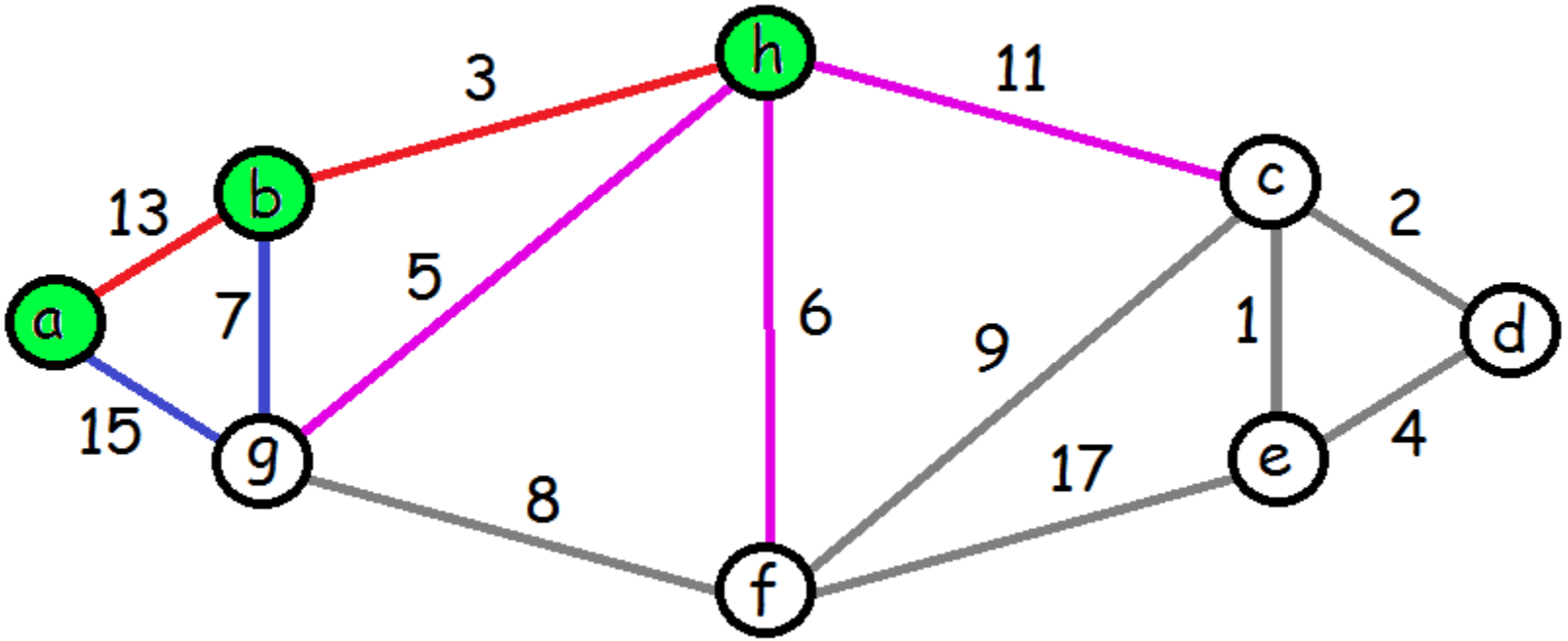
Green vertices: tree vertices

Purple edge: Minimum weight edge to a non-tree vertex from some tree vertex.

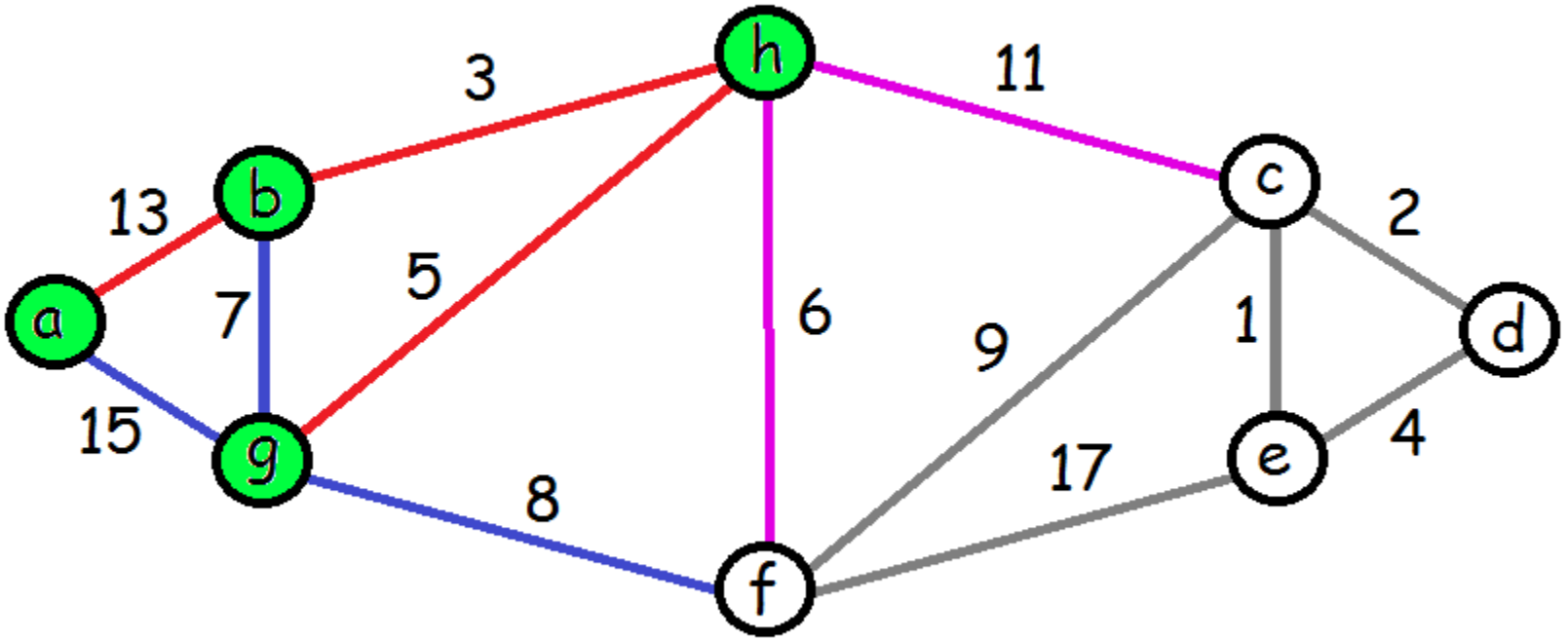
Dijkstra/Prim algorithm:



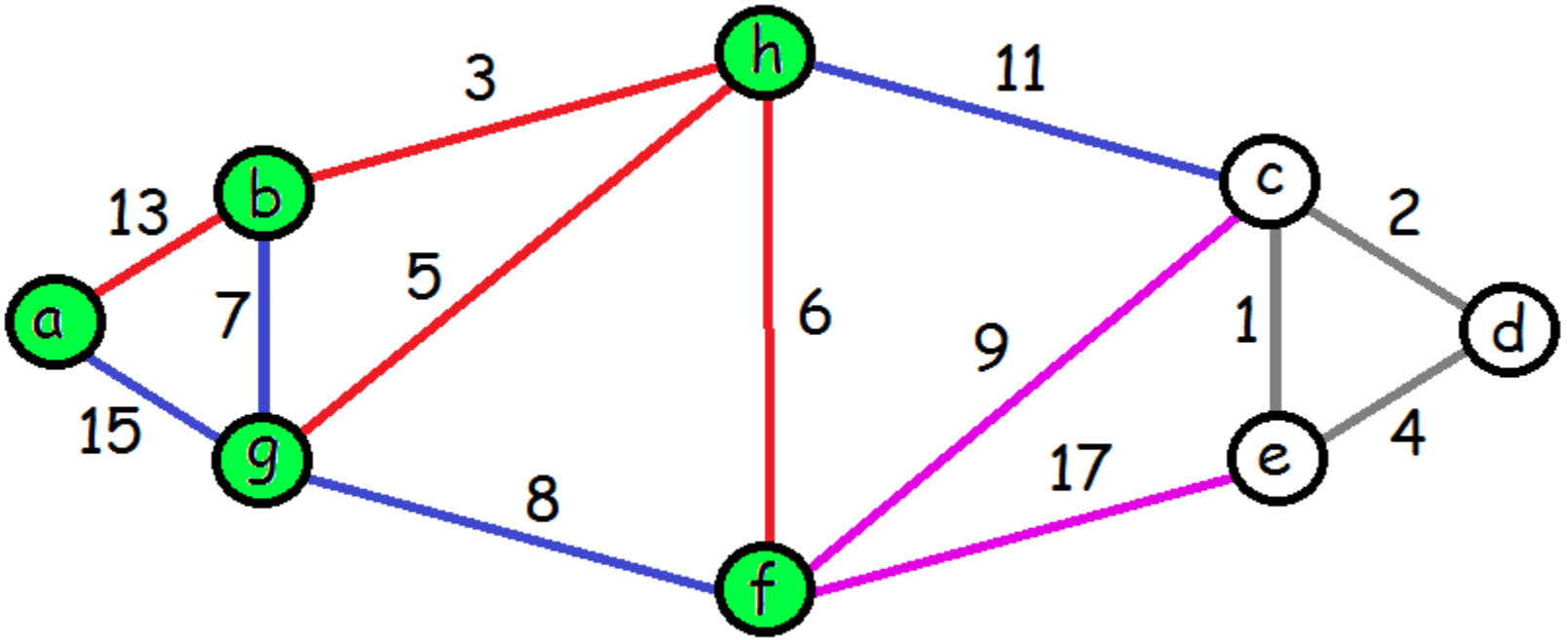
Dijkstra/Prim algorithm:



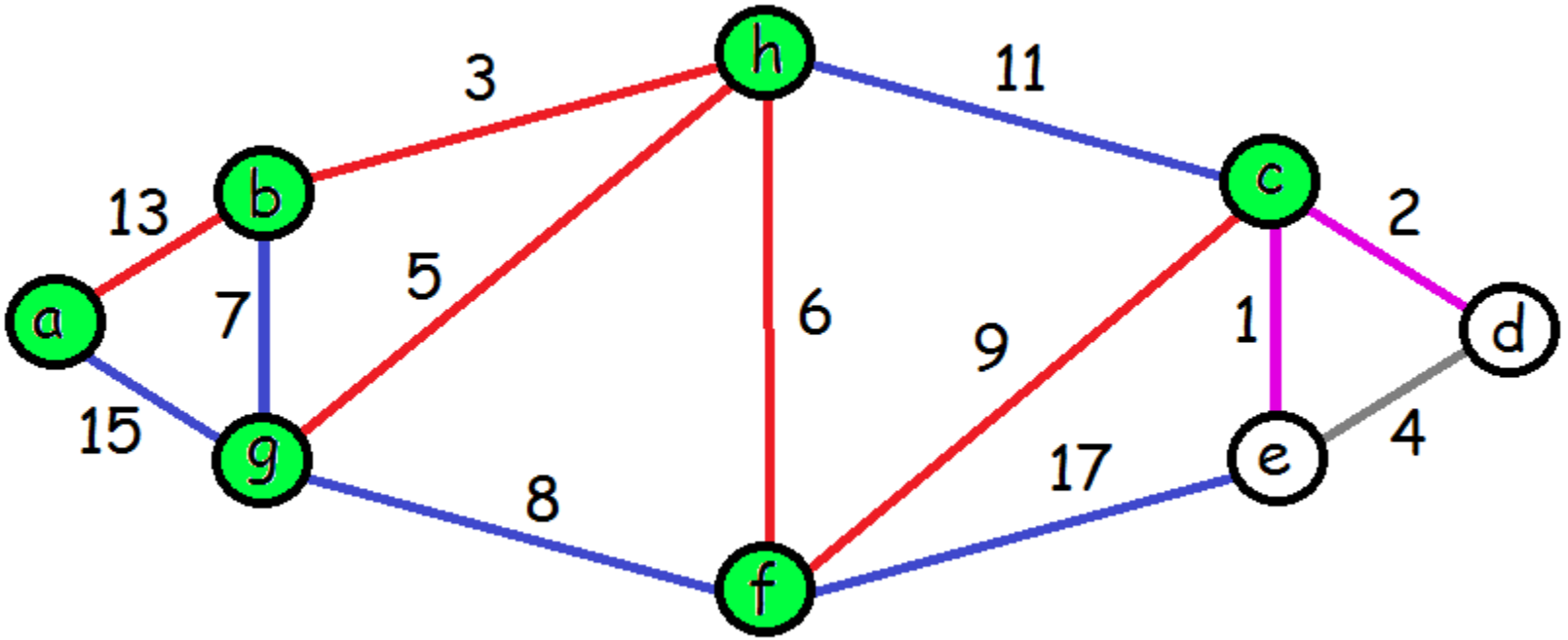
Dijkstra/Prim algorithm:



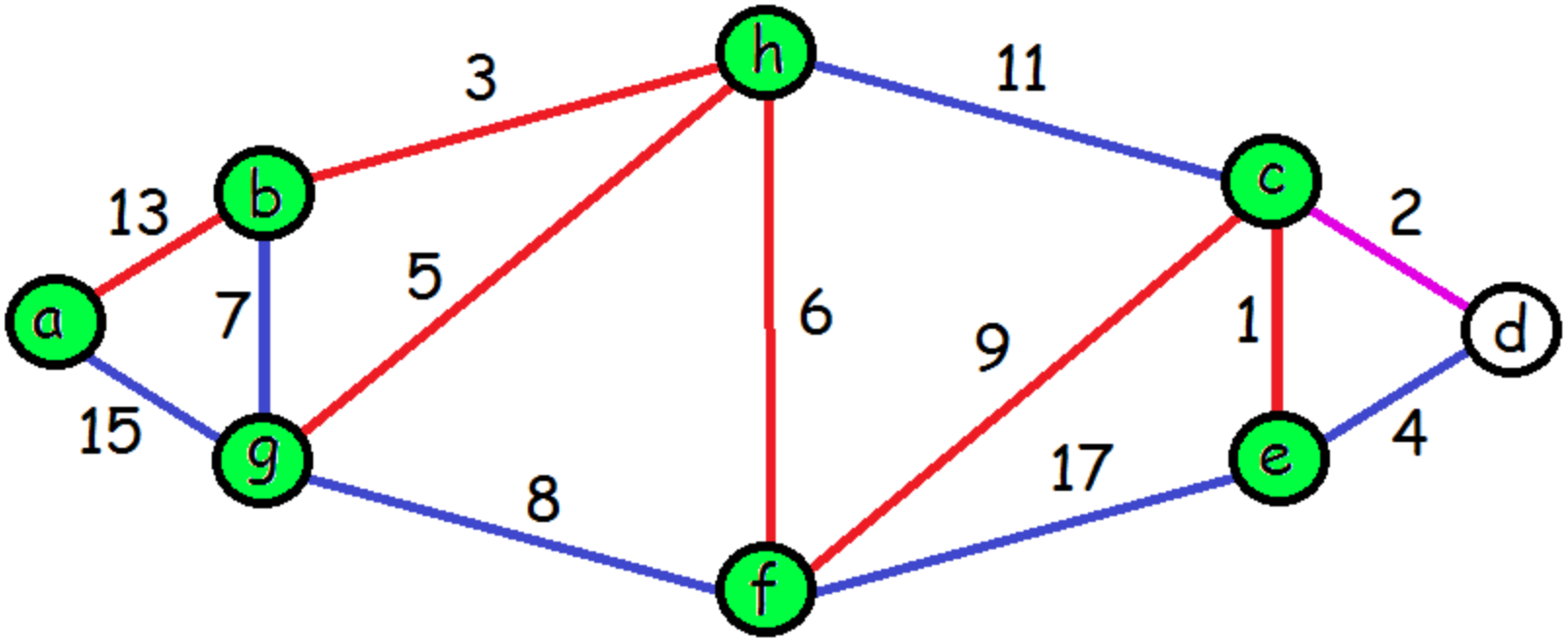
Dijkstra/Prim algorithm:



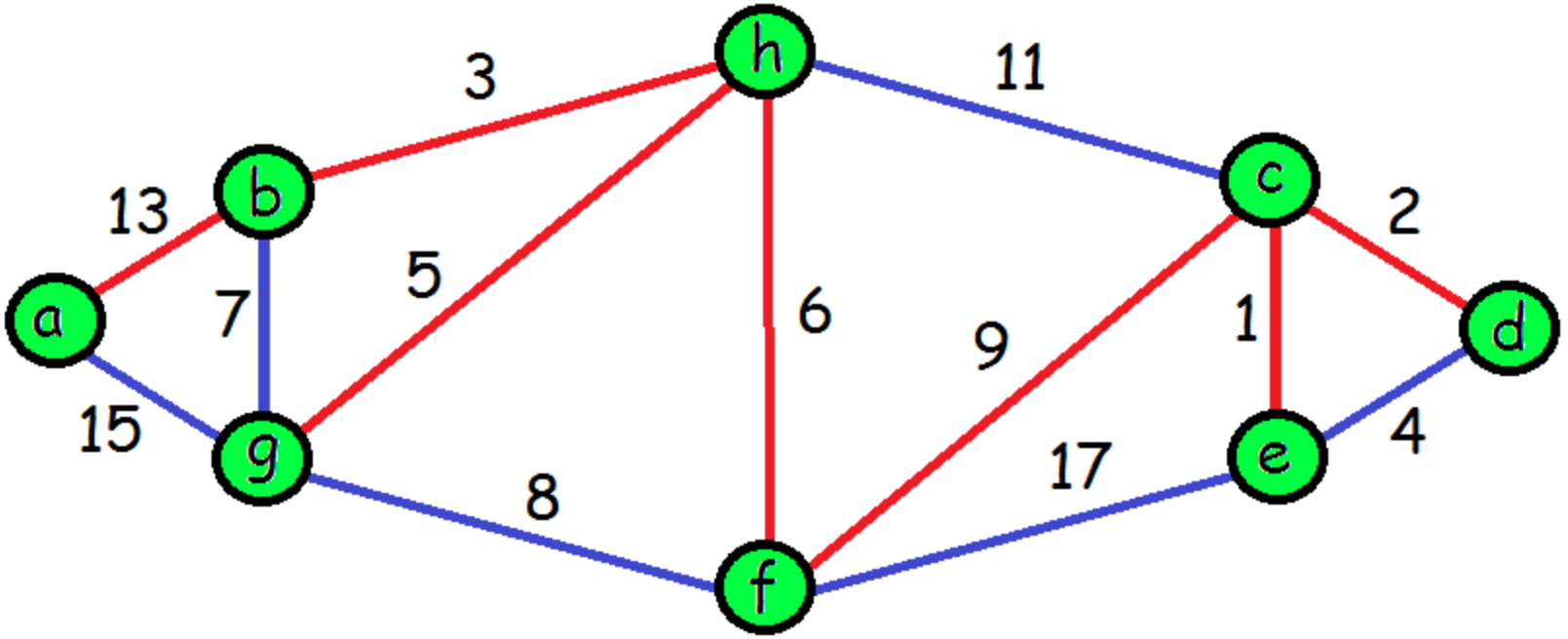
Dijkstra/Prim algorithm:



Dijkstra/Prim algorithm:

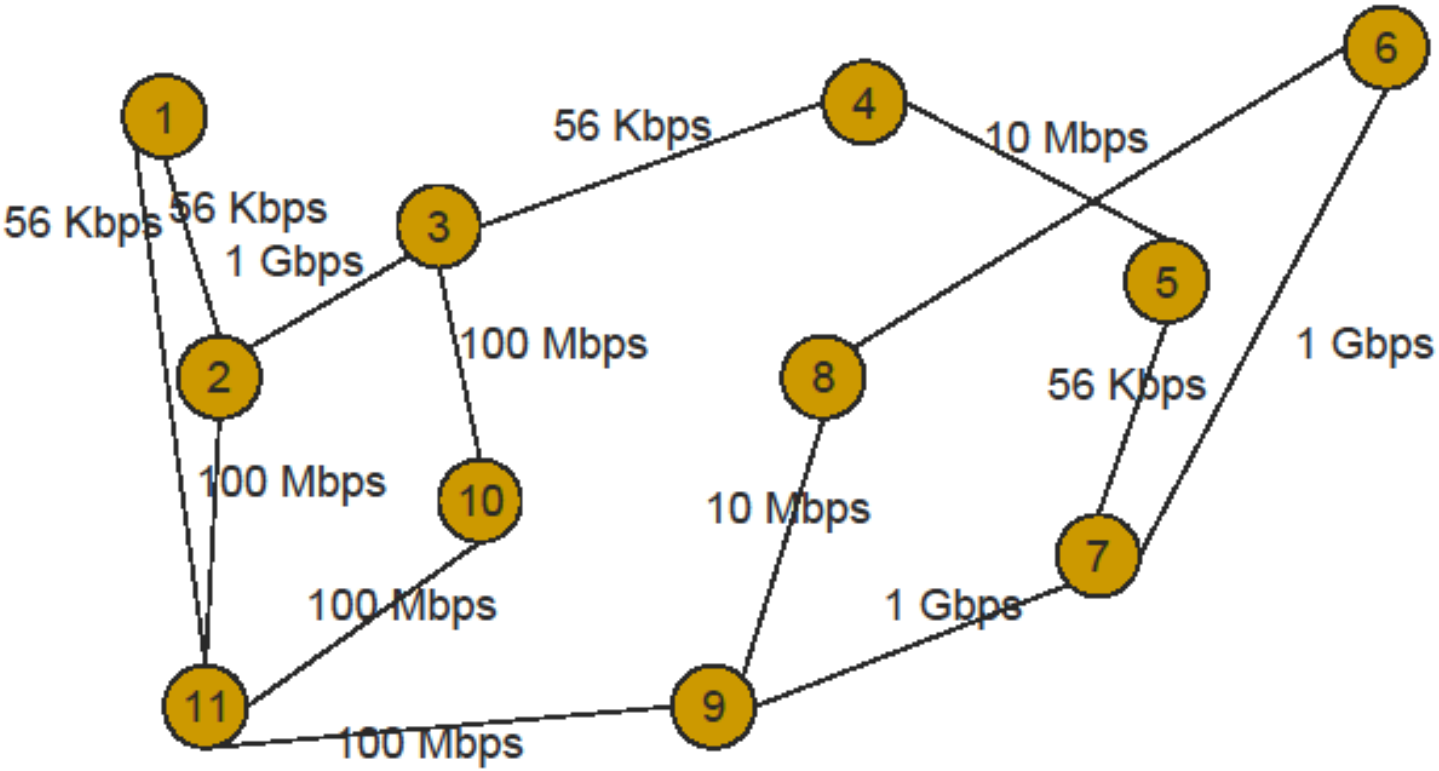


Dijkstra/Prim algorithm:



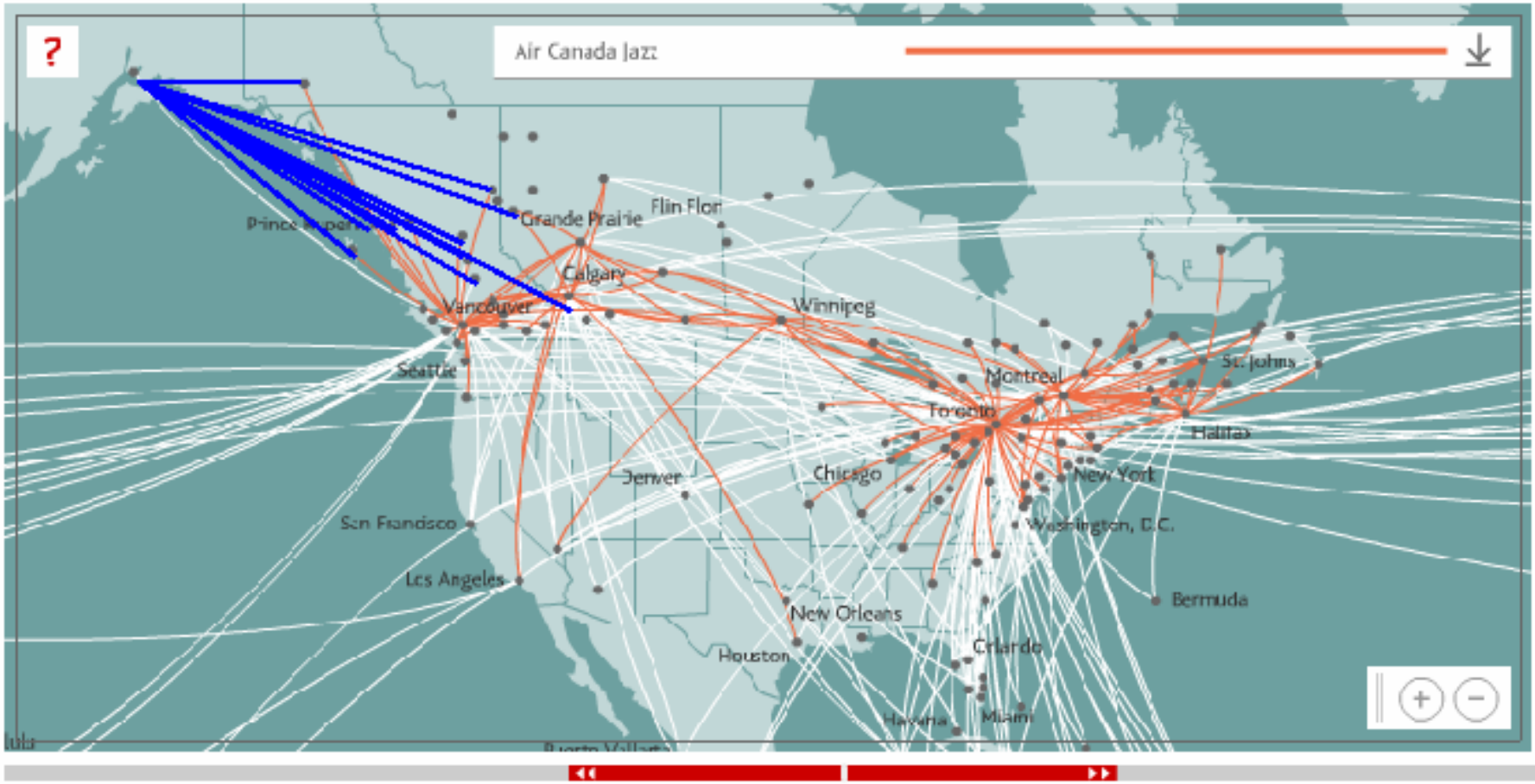
Communication Speeds in a Computer Network

Find fastest way to route a data packet between two computers



AIR CANADA

Select a map > World · Canada · United States · Europe · Asia / Pacific · Latin America



Exit Map

Dijkstra/Prim Algorithms

Use tree, min_wt, and closest.

Update to min_wt:

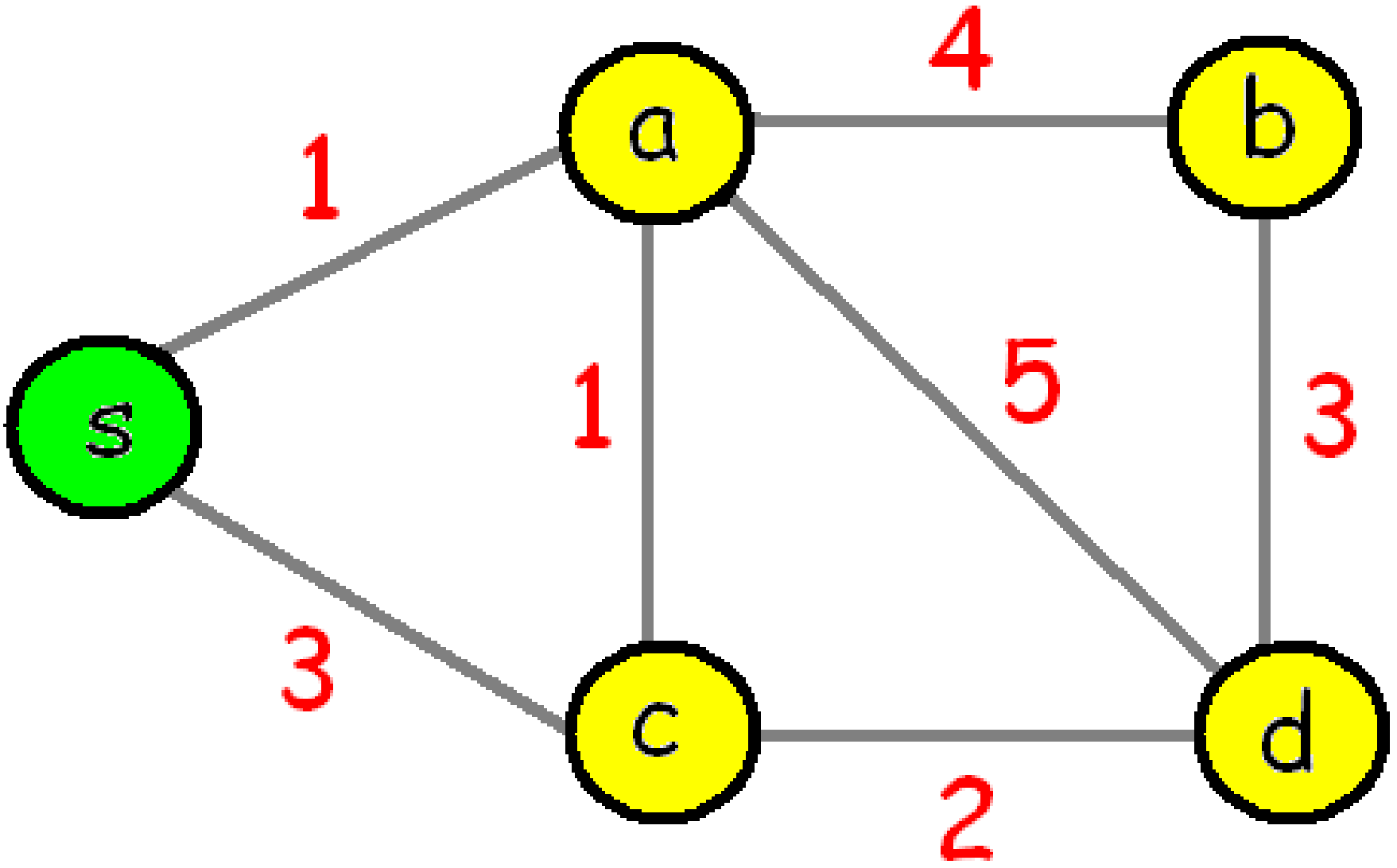
Adding v to the tree, neighbour is u.

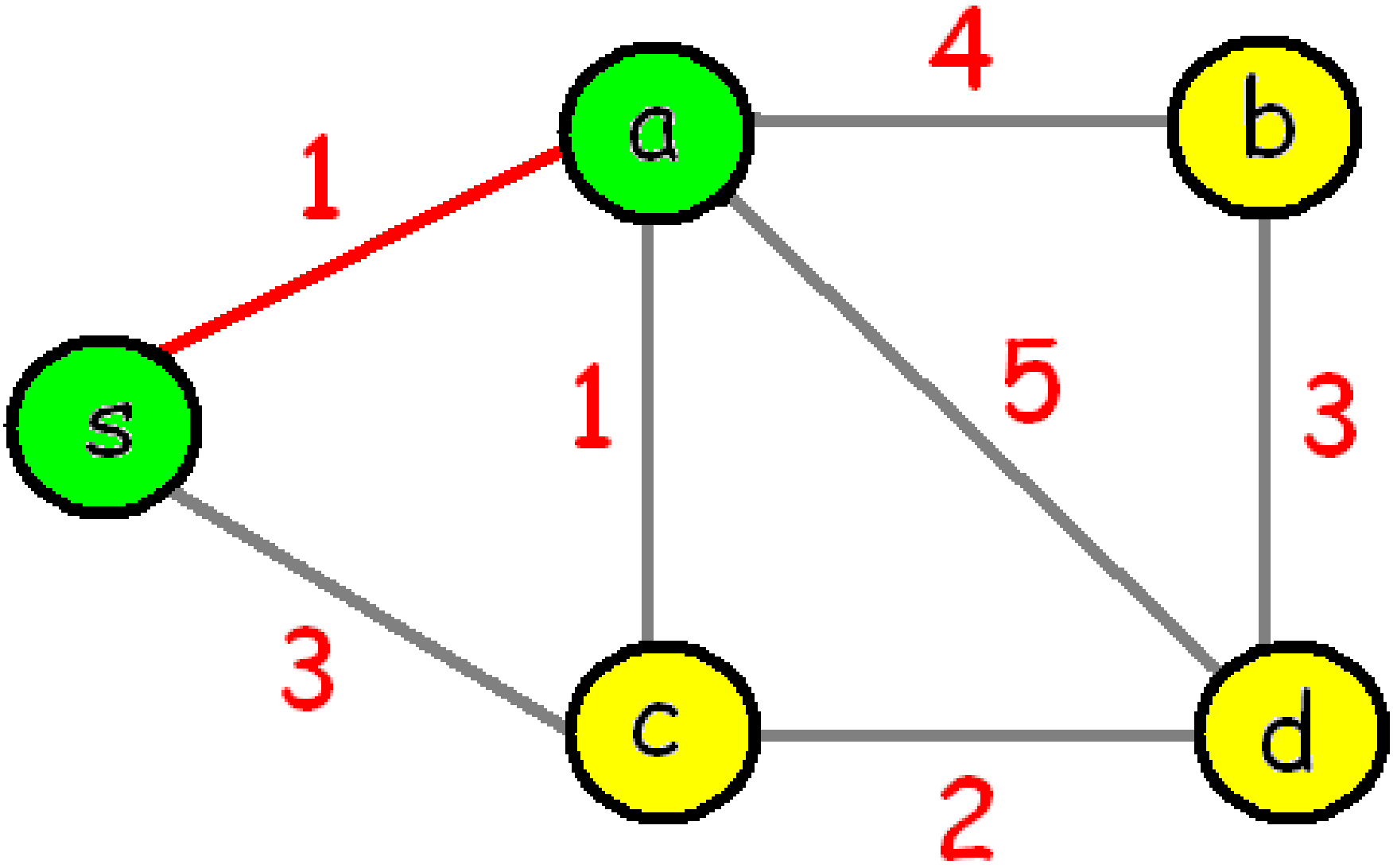
w= weight(u,v)

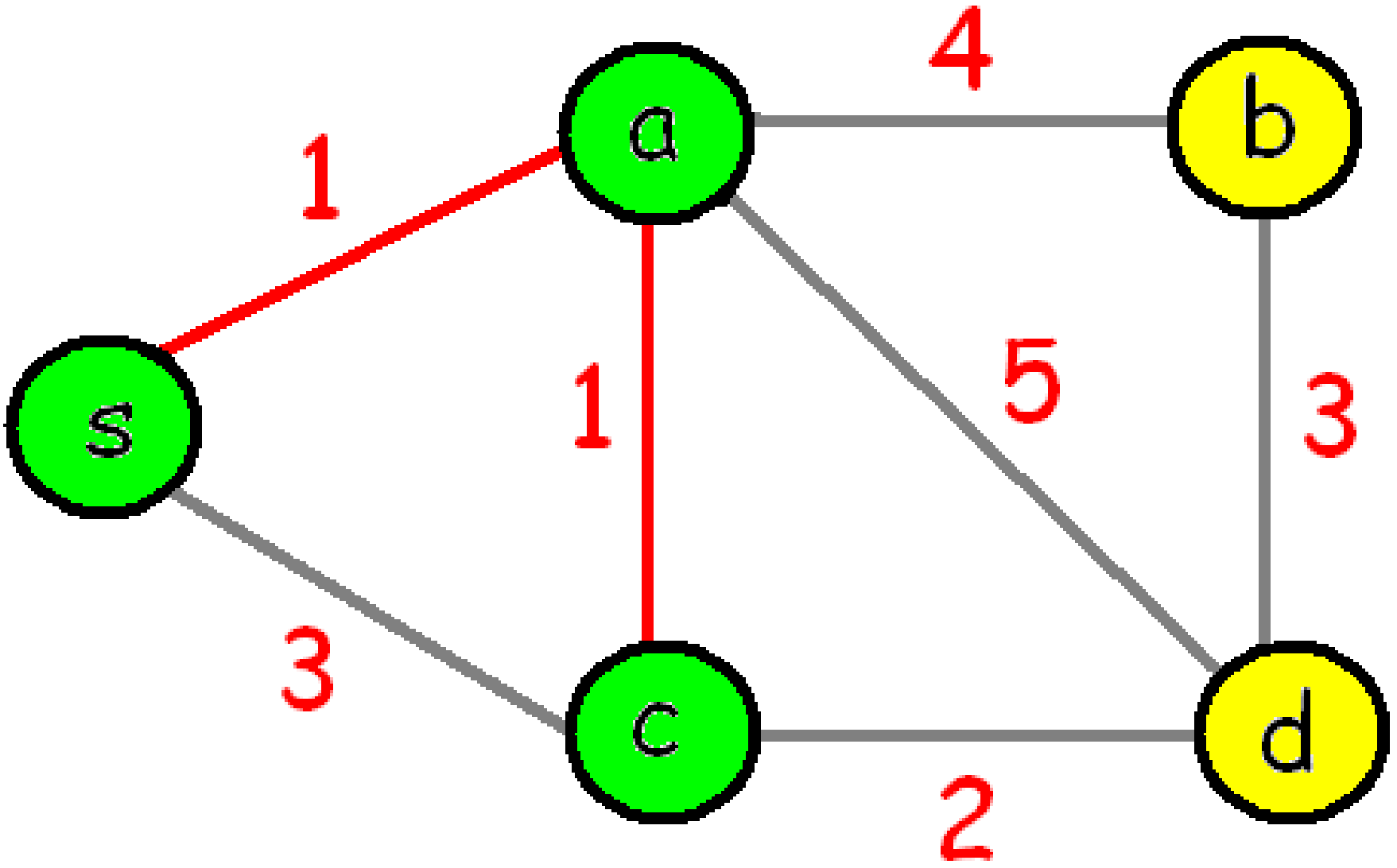
MST: if w < min_wt[u] then closest[u]=v and min_wt[u]=w

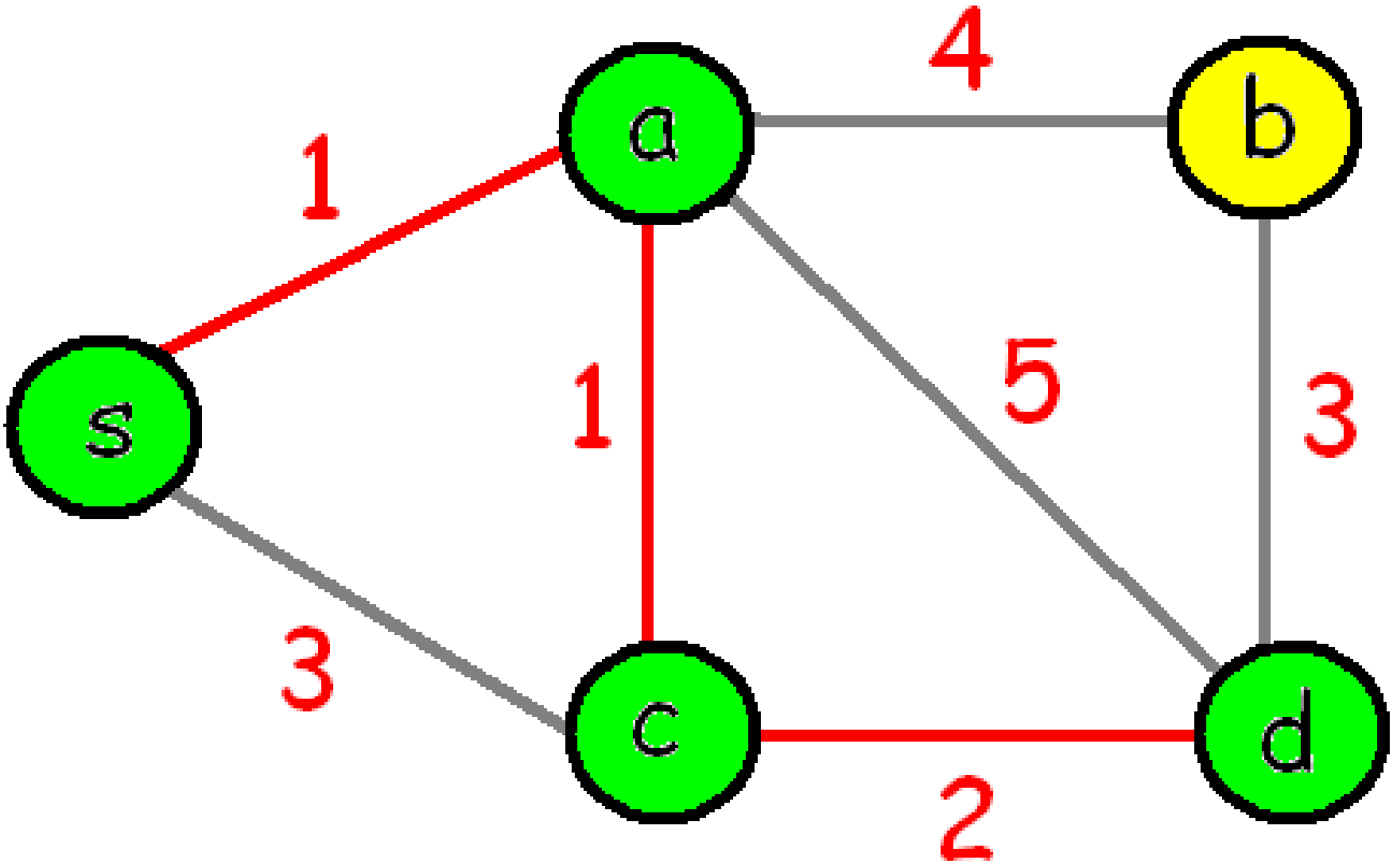
Shortest Path: min_wt[s]=0

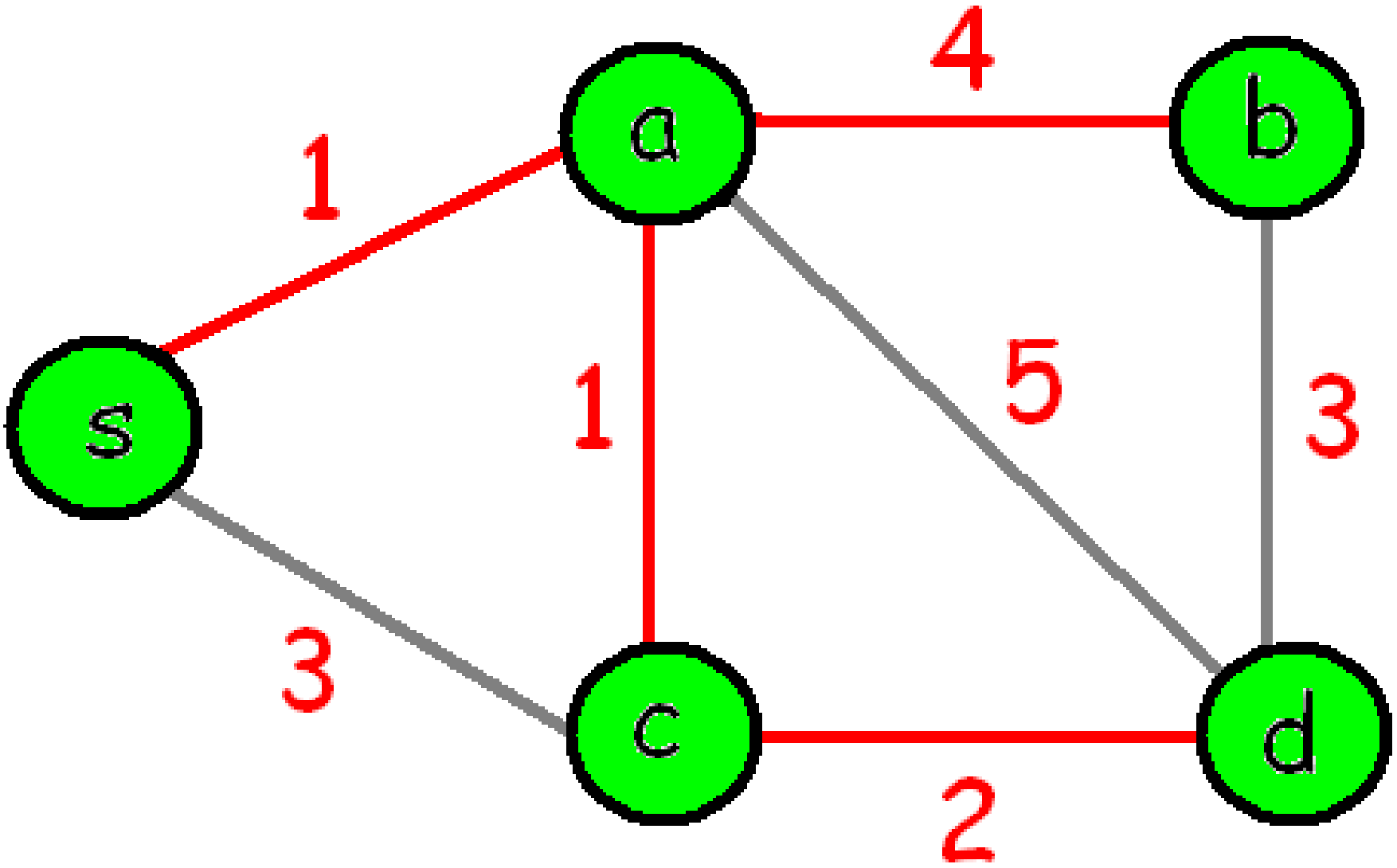
If min_wt[v] + w < min_wt[u] then closest[u]=v and min_wt[u]= min_wt[v]+ w


a
b
s
c
d
4
1
1
5
3
3
2


a
b
s
c
d
4
1
1
5
3
3
2


a
b
s
c
d
4
1
1
5
3
3
2


a
b
s
c
d
4
1
1
5
3
3
2


a
b
s
c
d
4
1
1
5
3
3
2

What is the time complexity of Dijkstra/Prim (MST or Shortest Path) if a min-heap is used to find the min weight at each phase, and Bubbleup is used to restore the heap when a weight is decreased?

Does this always make the algorithm faster?

Tutorials: What happens in them? Would you prefer more structured ones (posted problems to solve) or do you prefer to just ask questions with in trouble areas? How could they be structured to better help you to learn the class material?

Do you like our unusual grading scheme?

Should CSC 225 have more or less programming?

Do you feel you learned skills in this class or just memorized and regurgitated?