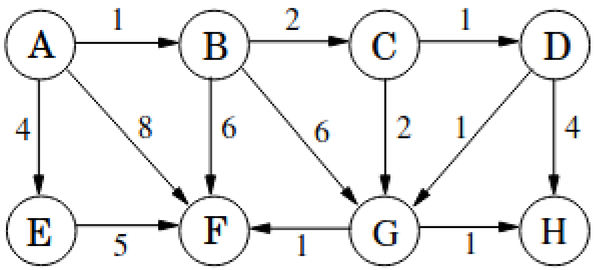
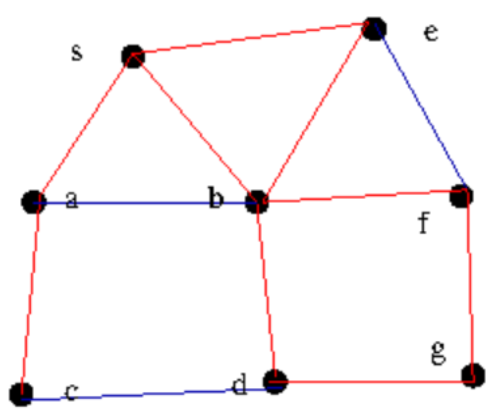
Tutorial 11.

1. Modify Dijkstra's algorithm to compute the number of shortest paths from s to every vertex t.
2. Compute the shortest path for all vertices starting from A. Do this in tabular form.



3. Show an example of a graph with negative edge weights and show how Dijkstra's algorithm may fail. Suppose that the minimum negative edge weight is -d. Suppose that we create a new graph G' with weights w', where G' has the same edges and vertices as G, but w'(e)=w(e)+d. In other words, we have added d to every edge weight so that all edges in the new graph have edge weights non-negative. Let us run Dijkstra on this graph. Will it return the shortest paths for G?
4. Look at the following graph from Tutorial 9 with red edges and blue edges. Our task was to find the path from s to every vertex t, with the fewest red edges. Run any modified bfs of your choice and Dijkstra and compare the sequence of vertices visited by BFS and by Dijkstra.



5. You are given a time table for a city. The city consists of n stops V={v1,v2,...,vn}. It runs m services s1,s2,...,sm. Each service is a sequence of vertices and timings. For example, the schedule for service K7 is given below. Now, you are at stop A at 8:00am and you would like to reach stop B at the earliest possible time. Assume that buses may be delayed by at most 45 seconds. Model the above problem as a shortest path problem. The answer should be a travel plan.

Service : K7			
H15	Convocation Hall	Market Gate	H15
7:15am	7:20am	7:30	7:40