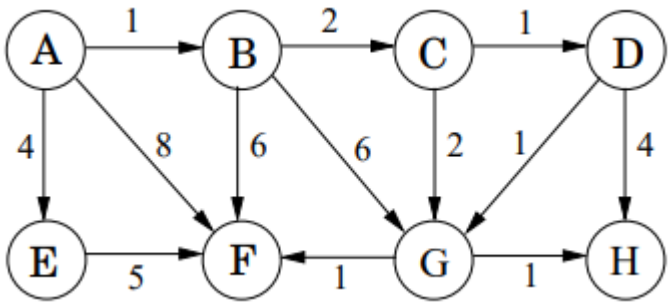
Homework 4
Design and Analysis of Algorithms
Sections BCS-4C, BCS-6A, Spring 2021

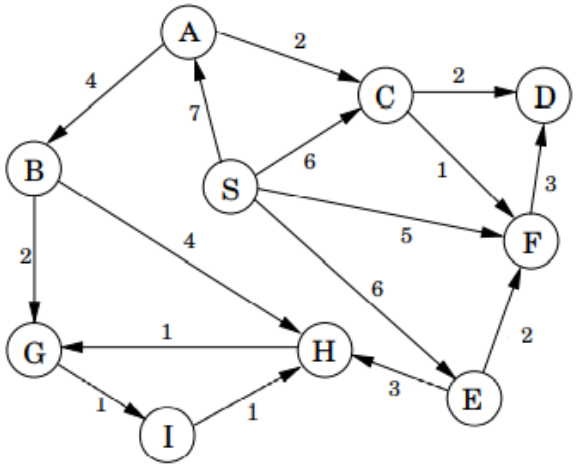
Due Thursday, June 10, 2021 at 1159pm

1. Dry run Dijkstra's Algorithm on Graph I and Graph II given below. Show the node extracted from the min heap in each iteration of the algorithm; also, show the distance of extracted node from the source and its parent node. If two or more nodes in the heap have the same distance value which is minimum, extract the one which is alphabetically first. Take node A as the source in Graph I and S as the source in Graph II.

   For example, in Graph I below, the first output should be: **[A, 0, nil]**, which means node A has distance 0 from the source and its parent is nil. Similarly, the second output is **[B, 1, A]**, and so on.

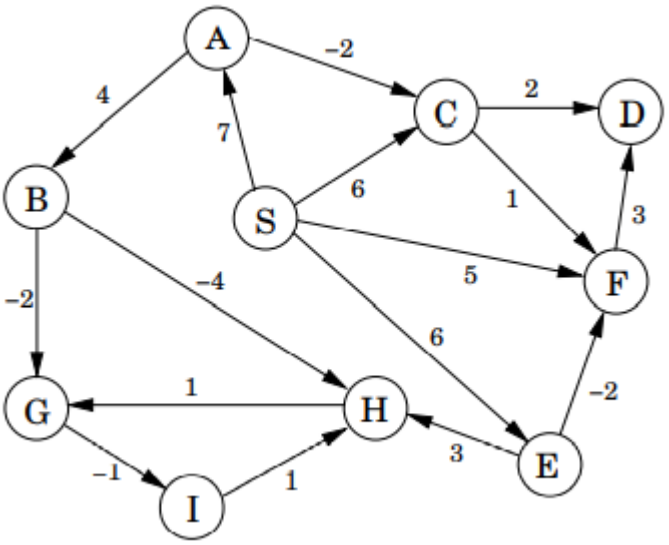


Graph I



Graph II

2. Dry run Bellman Ford's algorithm on Graph III given below. Take S as the source.



Graph III

In each phase, relax the edges in this order: (A, B), (A, C), (B, G), (B, H), (C, D), (C, F), (E, F), (E, H), (F, D), (G, I), (H, G), (I, H), (S, A), (S, C), (S, E), (S, F).

Show the distance and parent values of each node after every phase. You may do this in a tabular form, with one row per node and column showing the status of that node after the next phase. For instance, the first column in the table (before any phase has executed) will look like the following:

A	(0,nil)
B	(∞, nil)
C	(∞, nil)
D	(∞, nil)
E	(∞, nil)
F	(∞, nil)
G	(∞, nil)
H	(∞, nil)
I	(∞, nil)
S	(∞, nil)

How many phases does it take for the algorithm to find the SSSP tree?

3. Sometimes it is desirable not only to find the shortest path in a graph (assuming only positive weights here), but also among the shortest paths, the one that has the minimum number of edges. Modify Dijkstra's algorithm so that it makes sure that for each vertex t, the shortest path we get from s (source) to t, is the one with the minimum cost but also, among all the minimum cost paths, the one with the minimum number of edges. Take the pseudo-code from class and add your additional code to it. Also explain your logic in a few lines of English.

4. Imagine that you have been given the guarantee that no shortest path in your graph has more than 3 edges in it. You are then asked to find all the shortest paths. What is the fastest time in which you can find these paths? Explain your approach.

5. As the mayor of Billipura, you have been given the budget to make a new road in your district. You have a map of your district: a directed graph with positive edge weights, G= (V, E), w. You know, being an experienced politician, that in order to gain the most number of votes in the next elections, it matters most to decrease the distance between towns **s** and **t** (two specific nodes in G, given as inputs to the algorithm). The options for the new roads are present in an edge list E', where |E'| = k. You need to figure out which one of the k edges you should add to G, so that the distance between s and t is decreased the most. Give the most efficient algorithm you can for this problem. You may assume: $E \cap E' = \emptyset$.