CMSC351 - Fall 2014, Homework #6

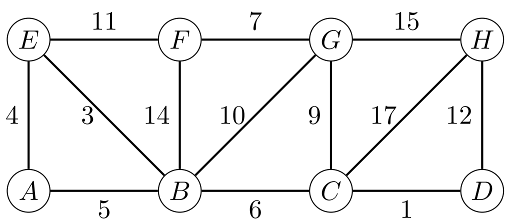
Due: December 12th at the start of class

PRINT Name:

- Grades depend on neatness and clarity.
- Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS prove the correctness of your algorithms either directly or by referring to a proof in the book.
- Write your answers in the spaces provided. If needed, attach other pages.
- The grades would be out of 100. Four problems would be selected and everyone grade would be based only on those problems. You will also get 25 bonus points for trying to solve all problems.

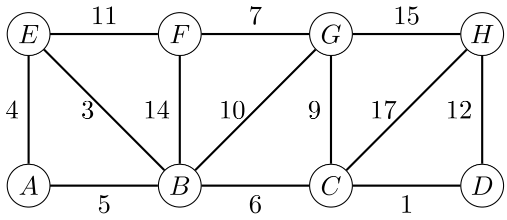
Note: In this problem set, n denotes the number of vertices and m denotes the number of edges. By a linear algorithm, we mean an algorithm that runs in $O(m+n)$.

Problem 1 Run Dijkstras algorithm on the weihgted graph below, using vertex A as the source. Write the vertices in the order which they are marked and compute all distances at each step.



Problem 2 (a) Consider the weighted graph below. Run Prim's algorithm starting from vertex A. Write the edges in the order which they are added to the minimum spanning tree.

(b) Consider the weighted graph below. Run Kruskal's algorithm starting from vertex A. Write the edges in the order which they are added to the minimum spanning tree.



Problem 3 **(CLRS 24.3-2)** Give an example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers. Write the vertices in the order which they are marked during Dijkstra's algorithm.

Problem 4 **(CLRS 24.3-4)** Professor Gaedel has written a program he claims implements Dijsktra's algorithm. The program produces the *distance* and the *parent* for each vertex v in the graph. Assume your are given the graph and the output of the professor's program, i.e., the distance and the parent for each vertex v. Design an $O(n+m)$ algorithm to determine whether the distance and the parent attributes match those of some shortest-path tree. You may assume that all edge weights are non-negative.

Problem 5 **(CLRS 22.3-7)** Let $G = (V, E)$ be a weighted graph and T be its shortest-path tree from source s. Assume all weights in G are increased by the same amount, i.e., $\forall e \in E, w'_e = w_e + c$. Is tree T still the shortest-path tree (from source s) of the modified graph? If yes, prove the statement. Otherwise, give a counter example.

Problem 6 BELLMAN-FORD's algorithm computes all shortest paths from a single source s when the input graph has no negative-weight cycle (The weight of a cycle $C = (v_0, v_1, \cdots, v_t)$ is defined as $w_{v_t v_0} + \sum_{i=1}^{t} w_{v_{i-1}, v_i}$). This algorithm works when the input graph has non-negative weights. Here is the pseudo-code for BELLMAN-FORD's algorithm:

Algorithm 1 BELLMAN-FORD(s)

```
1: for i ← 1 to n do
2:     dist[i] = ∞
3: end for
4: dist[s] = 0
5: for i ← 1 to n − 1 do
6:     for all edges such as (u, v) do
7:         if dist[v] > dist[u] + w_uv then
8:             dist[v] = dist[u] + w_uv
9:             parent[v] = u
10:        end if
11:    end for
12: end for
```

(a) What is the running time of the algorithm?

(b) Consider an arbitrary vertex v and let $s = v_0, v_1, \cdots, v_k = v$ be the shortest path from s to v. Define $f(v) = k$ i.e, the length of the shortest path from s to v. What is $f(s)$?

(c) Prove $\forall v, 0 \leq f(v) \leq n - 1$.

(d) Prove by induction on $f(v)$ that the algorithm will compute the shortest path for vertex v when $i = f(v)$ (line 5-12 of the pseudo-code)?

(e) Give an example of a graph with a negative-weight cycle for which BELLMAN-FROD's algorithm produces incorrect answers.

Problem 7 Let D be the shortest path matrix of weighted graph G. It means $D[u, v]$ is the length of the shortest path from vertex u to vertex v, for every two vertices u and v. Graph G and matrix D are given. Assume the weight of an edge e is decreased from w_e to w'_e. Design an algorithm to update matrix D with respect to this change. The running time of your algorithm should be $O(n^2)$. Describe all details and write a pseudo-code for your algorithm.

Problem 8 Let $G = (V, E)$ be a connected weighted graph, and let T be a minimum spanning tree of G. Graph G and tree T are given. Assume the cost of one edge e in G is decreased from w_e to w'_e. Design an algorithm to find a minimum spanning tree in the modified graph. The running time of your algorithm should be $O(n)$. Describe all details and write a pseudo-code for your algorithm.