Homework No. 5

CSCI 4470/6470 Algorithms, CS@UGA, Spring 2017

Due Thursday April 6, 2017

The answers must be word-processed or typed. You may substitute formulae and figures with hand-writings. Your submitted algorithms should be in the pseudo-code, not in any specific programming language. Answers deviating from these requirements will be returned without grading.

The answers must be the student’s own work. Idea sharing and referencing to others’ work are not allowed. Plagiarism and other forms of academic dishonesty will be handled within the guidelines of the Student Handbook and reported to the University.

120 points in total (including 20 bonus points for undergraduates).

1. (20 points) Let $G = (V,E)$ be a weighted graph (with edge weight function $w$). Assume that edges in $G$ have unique weights. Let $(u,v)$ and $(x,y)$ be the two edges with the two least weights, and $w(u,v) < w(x,y)$.

   (1) Prove that every minimum spanning tree has to contain edge $(u,v)$.

   (2) Does every minimum spanning tree have to contain edge $(x,y)$ as well? Explain.

   Note that this question has nothing to do with how the tree $T$ was generated.

2. (15 points) Let $G = (V,E)$ be a weighted graph (with edge weight function $w$) and $T \subseteq E$ be a minimum spanning tree of $G$. Prove that for every edge $(u,v) \in T$, there is a cut in graph $G$ such that $(u,v)$ is a light edge crossing the cut. Note that this question has nothing to do with how the tree $T$ was generated.

3. (15 points) (1) How does algorithm MST-Kruskal identify a cut and pick a light edge crossing the cut to add to the set $A$?
(2) How does algorithm MST-Prim store $A$? How does the algorithm ensure that the edge newly added to set $A$ does not form a cycle with those edges already in $A$?

4. **(20 points)** (1) Run the Bellman-Ford algorithm on the directed graph of Figure 1, using vertex 4 as the source. Show $d$ and $\pi$ values after each of the $|V| - 1$ passes.

(2) Modify the graph so that it contains at least one negative cycle. Run the Bellman-Ford algorithm again to show how the negative cycle can be detected by the algorithm.

![Figure 1: Graph for Question 4.](image)

5. **(15 points)** Bellman-Ford algorithm was designed to be able to handle graphs that may contain negative cycles. When the input graph is a DAG (directed acyclic graph), would the algorithm use the same amount of time as algorithm DAG-Shortest Paths? Explain.

6. **(15 points)** Give a simple example of a directed graph with negative weight edges for which Dijkstra’s algorithm produces incorrect answers. Also argue that, if all negative weights occur only on the edges leaving the source $s$, Dijkstra’s algorithm remains correct.

7. **(20 points, Graduate student only, bonus for undergraduates)** How do we use Floyd-Warshall algorithm to detect if the input graph contains a negative weight cycle?