Fundamental Algorithms
Courant, Fall 2017

Homework 11

Instructor: Amir Shpilka
Due Date: Wed Dec 6, 11:55pm

1. (a) (4 points) Assume that all edge weights of an undirected graph G are equal to the same number w. Design the fastest algorithm you can to compute the MST of G. Argue the correctness of the algorithm and state its run-time. Is it faster than the standard $O(E + V \log V)$ run-time of Prim?

   (b) (6 points) Now assume the all the edge weights are equal to w, except for a single edge $e' = (u', v')$ whose weight is w' (note, w' might be either larger or smaller than w). Show how to modify your solution in part (a) to compute the MST of G. What is the running time of your algorithm and how does it compare to the run-time you obtained in part (a) (or standard Prim)?

2. (8 points) Assume all edge weights of a connected undirected graph G are integers from 1 to W. Show how to modify Prim's algorithm to achieve running time $O(E + VW)$. Hence, if $W = O(1)$, you get the optimal time $O(E + V)$.

3. (12 points) To implement Kruskal's algorithm, we need a disjoint-set data structure that can perform the operations MAKE-SET, FIND-SET and UNION. To get Kruskal's algorithm to run in time $O(E \log V)$, we need this disjoint-set implementation to have the following property: any sequence of m operations runs in $O(m \log n)$ time, where n is the number of MAKE-SET operations. Describe an implement of the disjoint-set data structure with the required running time, and prove it's correctness and running time.

   Hint: if you take two trees such that each tree has at least 2^h nodes (where h is the height of the tree) and connect the root of the shallower tree to that of the deeper tree, then the resulting tree also has at least 2^h nodes (where h is the height of the resulting tree).

4. (5 points) Run the Bellman-Ford algorithm on the directed graph shown below, using vertex z as the source. In each pass, relax edges in this order: $(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)$ and show the d and π values after each pass. Now, change the weight of edge (z, x) to 4 and run the algorithm again, using s as the source.