CS 5633 Analysis of Algorithms – Spring 10

3/30/10

8. Homework

Due **Tuesday 4/13/10** before class

1. **Kruskal's tree (4 points)**
   Since Kruskal's algorithm computes a forest of trees which are incrementally merged, it is not immediately obvious why the output is actually a tree.

   Argue why Kruskal's algorithm computes a tree (= connected acyclic graph). Do not use the fact that somebody else has proven that Kruskal's algorithm computes an MST.

2. **Negative edge weights (2+2+4 points)**
   a) Give an example of a directed connected graph with real edge weights (that may be negative) for which Dijkstra's algorithm produces incorrect answers. Justify your answer.
   b) Does Dijkstra's algorithm only produce incorrect answers in the presence of a negative weight cycle, or could it also produce incorrect answers in the mere presence of negative weight edges (without any negative weight cycles)? Justify your answer.
   c) Suppose the weighted, directed graph $G = (V, E)$ has a special structure in which edges that leave the source vertex s may have negative weights. All other edge weights are nonnegative, and there are no negative-weight cycles. Show that Dijkstra's algorithm correctly finds shortest paths from s in G.

3. **Adding an edge (4 points)**
   Let $G = (V, E)$ be a connected undirected graph with weight function $w : E \to \mathbb{R}_0^+$ (i.e., all edge weights are ≥ 0). Assume edge weights are distinct. Further, let a minimum spanning tree T on G be given.

   Now, assume that one new edge (u, v), with $u, v \in V$, with weight $w(u, v)$ is added to G. (This weight is different from all other edge weights.)

   Give an efficient algorithm to test if T remains the minimum spanning tree for this new graph. Your algorithm should run in $O(|E|)$ time. Can you make it run in $O(|V|)$ time?

4. **Faster MST (8 points)**
   Let $G = (V, E)$ be a connected undirected graph with edge weights $w : E \to \mathbb{R}$.

   (a) If all of the edge weights are integers between 1 and $|V|$, how fast can the minimum spanning tree be computed? (Give the *most efficient* algorithm you can think of.)

   (b) Now assume all edge weights are integers between 1 and 10. Show that Prim's algorithm can be implemented to work in $O(|V| + |E|)$ time in this case. *(Hint: Suggest a data structure that replaces the priority queue and analyze the runtime.)*