CMPS 6610 Algorithms – Fall 16

11/8/16

7. Homework

Due **11/15/16** at the beginning of class

1. **Union by Weight (3 points)**
   Consider a disjoint-set forest implementation that uses union by weight only. Give a sequence of m MAKE-SET, UNION, and FIND-SET operations, n of which are MAKE-SET operations, that takes $\Omega(m \log n)$ time.

2. **Link Before Find (5 points)**
   In a disjoint-set forest implementation, the UNION operation is implemented as UNION(x, y)=LINK(FIND-SET(x), FIND-SET(y)).

   (a) Consider a disjoint-set forest implementation that uses both union by weight and path compression. Show that any sequence of m MAKE-SET, FIND-SET, and LINK operations, where all the LINK operations appear before any of the FIND-SET operations, takes only $O(m)$ time.
   Hint: Use amortized analysis, possibly the accounting method.

   (b) What happens in the same situation if we use path-compression only (and not union by weight)?

3. **Dijkstra variants (8 points)**

   (a) Let G be a directed graph with non-negative edge weights. Consider changing the condition in the while loop of Dijkstra's algorithm to $|Q| > 1$. Prove that at the end of the algorithm the d-values contain the shortest path weights (so, the shortest path weights are correctly computed even when skipping the last iteration from Dijkstra's algorithm).

   (b) Let G be a directed graph with non-negative edge weights. Consider changing the condition in the while loop of Dijkstra's algorithm to $|Q| > 2$. Show that at the end of the algorithm the d-values do not necessarily contain the shortest path weights (give a counter-example).

   (c) Now, let G be a directed graph with arbitrary edge weights, and assume that some of the edge weights are negative. Let W be the smallest (negative) weight of any edge. Consider reweighing each edge weight by adding $-W$ to every edge, which yields a graph G' with non-negative edge weights. Then run Dijkstra's algorithm on G', and in the end add W to every computed d-value. Give a counter-example to show that this approach does not always compute the correct d-values.

FLIP OVER TO BACK PAGE $\Longrightarrow$

4. **Negative edge weights (5 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) 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.

5. **Negative-weight cycle (4 points)**
   Given a directed weighted connected graph $G = (V, E)$ with real edge weights (i.e., negative edge weights are allowed). Give an algorithm that detects **and prints** out a negative-weight cycle if G contains a negative-weight cycle. (Do not use Floyd-Warshall's algorithm.) What is the runtime of your algorithm?