CMPS 4610 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 (4 points)**
   In a disjoint-set forest implementation, the Union operation is implemented as Union($x, y$)=Link(Find-Set($x$), Find-Set($y$)).

   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.*

3. **Dijkstra variants (5 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).

4. **Negative edge weights (3 points)**

   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?