Nigel Decontie
V00853112

CSC 226 Assignment 3

1. Adding two trees of different heights will leave the overall height unchanged. Adding two trees of equal height will increase the overall height by 1.
   The size of the tree is at least 2^h where h is the height of the tree.
   Base case: h = 0
      $2^0 = 1$, which is true for a graph of one node.
      $2^1 = 2$, which is true since there is one parent and one child.
   Any additional nodes can increase the size of the tree but not necessarily the height (ie. new node's parent already has children).
   As a result, we get the inequality:
      $2^h <= N$
      $lg(2^h) = lg(N)$
      $h = lg(N)$
   Therefore the height is at most lg(N).

2. Algorithm 4.10 uses topological sort, so every edge v->w will be relaxed at least once. When v is relaxed, the inequality distTo[w] <= distTo[v] + e.weight() is given. This inequality should always hold until the algorithm is done since distTo[v] doesn't change.
   Assume the inequality doesn't hold. That would mean that distTo[v] changed and has become smaller. However, since it is topologically sorted, no edge pointing to v should be processed after v is relaxed, unless there is a back edge to v. This would mean there is a cycle and the graph would not be a DAG. This is a contradiction however, since the algorithm only processes acyclic, digraphs.

3. If the PQ is unsorted, when relaxing edges, updating the edges in the PQ would require the algorithm to iterate through the PQ to find edges that need to be updated. This would have to be done for each edge.

4. V-1 iterations are required in the Bellman-Ford algorithm. If there is still an edge that can be relaxed then another walk around the cycle will reduce the cost. This suggests that by continuously going through this cycle, the cost can be reduced infinitely, resulting in no correct answer being determined.