Problem Set 5

CS 5633 – Spring 2015
Tom Bylander, Instructor

assigned March 26, 2015
first due date April 9, 2015
second due date April 16, 2015

1. (10 pts.) Suppose $x_1, x_2, \ldots, x_{12}$ are elements. Show the final data structure and the answers returned by the FIND-SET operations in the following program. Use the linked-list representation with the weighted-union heuristic. Assume that if the sets containing $x_i$ and $x_j$ have the same size, then the operation UNION($x_i, x_j$) appends $x_j$’s list onto $x_i$’s list.

   for $i \leftarrow 1$ to 12
      do MAKE-SET($x_i$)
   for $i \leftarrow 1$ to 8
      do UNION($x_i, x_{i+4}$)
   UNION($x_9, x_{10}$)
   UNION($x_{11}, x_{12}$)
   UNION($x_1, x_4$)
   FIND-SET($x_{12}$)
   FIND-SET($x_9$)

2. (10 pts.) Show the data structure and the answers returned by the FIND-SET operations in the above program. Use the disjoint-set forest representation with union by rank and path compression. Assume that if the sets containing $x_i$ and $x_j$ have the same rank, then the operation UNION($x_i, x_j$) results in the representative of $x_j$’s set becoming the representative of the union.

3. (10 pts.) Consider the following dag.

   ![Diagram of a dag](image-url)

   a -> b -> c -> d
   e -> j -> g
   h -> i -> j -> k
   l -> m -> n
Show how depth-first search works on the above graph. Assume that the second for loop of the DFS procedure considers the vertices in alphabetical order, and assume that the adjacency lists are ordered alphabetically. Show the discovery and finishing times for each vertex.

4. (10 pts.) Show the ordering produced by `TOPOLOGICAL-SORT` using the DFS from the previous exercise.

5. (10 pts.) Suppose \((u, v)\) is the minimum-weight edge incident on \(u\) in a graph \(G\), where \(G\) is undirected, connected, and weighted. Show that \((u, v)\) belongs to some minimum spanning tree of \(G\).

6. (10 pts.) Do Exercise 24.4-1 as follows. Show the corresponding constraint graph (see Figure 24.8 for an example). Show how the Bellman-Ford algorithm solves this problem. Assume \(x_1\) through \(x_6\) are initialized to 0. Show how the values change. Assume that the order of edges follows the order of difference constraints.

7. (10 pts.) In pseudocode, describe an algorithm to find the longest path in a dag. You may use the procedures defined in the book as subroutines.

8. (10 pts.) Consider the Knapsack problem. There are \(n\) items. The \(i\)th item is worth \(v_i\) dollars and weighs \(w_i\) pounds, both \(v_i\) and \(w_i\) are positive integers. We want to find the subset of items of maximum value, under the condition that the total weight of the subset cannot exceed \(W\).

Define the following function recursively:
\[
\text{opt}(i, w) = \text{the maximum value of any subset of the first } i \text{ items that weighs } w.
\]

9. (10 pts.) Continuing the previous exercise, write an algorithm in pseudocode that efficiently finds the maximum value of any subset that weighs \(W\) or less.

10. (10 pts.) Continuing the previous exercises, suppose that the values of \(\text{opt}(i, w)\) are stored in a table. Write an algorithm in pseudocode that inputs \(n, W\) and the \(\text{opt}(i, w)\) table and returns the optional subset.