CS381 **Homework 6** **Fall 2012**

Question 1. (20 points) Consider the following problem $\mathcal{P}$: Given an unsorted linked list L of n arbitrary distinct items, and an n-node binary tree T in which every internal node has exactly 2 children, the problem is to place the n items of L in the n nodes of T (one per node) such that for every internal node v whose left child is u and whose right child is w, the item at node v is greater than all the items in the subtree of u, and smaller than all the items in the subtree of w. Prove an $\Omega(n \log n)$ time lower bound for problem $\mathcal{P}$.

Question 2. (20 points) Given two unsorted lists A and B of respective lengths n_A and n_B, give an algorithm that computes the intersection of A and B in time $O((n_A + n_B) \log \min(n_A, n_B))$.

Question 3. (25 points) Let G be a connected undirected graph with n vertices and e edges. The graph G is already known to have no odd-length cycles in it (where the length of a cycle is the number of edges on it, i.e., every edge has a cost of 1). Give an $O(n^2)$ time algorithm for computing the length of a shortest cycle in G. (Note: An $O(ne)$ time algorithm is not acceptable.)

Hint. For every vertex v do a *breadth*-first search starting at v and stop that search as soon as you encounter a non-tree edge (one of these n "search-stopping" non-tree edges belongs to a shortest cycle).

Question 4. (25 points) Let G be a directed graph whose vertex set is $\{a, b, c, d, e, f, g, h, i, j, k, l, m\}$ and whose adjacency lists representation is given below.

```
L[a]: b, c
L[b]: a, d, e
L[c]: h, i
L[d]: e, g
L[e]: f, g
L[f]: d
L[g]: f
L[h]: d, j, l
L[i]: k, m
L[j]: d, h
L[k]: c, g
L[l]: j
L[m]: k
```

For example, the $L[a]$ list encodes the fact that vertex a is the tail of two directed edges (a, b) and (a, c). In answering the questions below, the order of the contents of each of the above lists is important (a different order for a list's contents will result in a different answer, so please use the above orders for list contents).

1. (10 points) Draw the depth-first search tree of G that results from carrying out a depth-first search starting from from vertex a. In the figure you draw, show all the tree edges, and write next to each vertex both its original name and its depth-first number.
2. (5 points) List the non-tree edges that are forward edges, those that are backward edges, and those that are cross edges; within each of these 3 categories of non-tree edges, the order in which you list them should be the same as the order in which they are encountered by the depth-first search.
3. (10 points) List the strongly connected components of G in the order in which they are produced by the algorithm we covered in class (within a component you can list the vertices in any order you want).

Question 5. (10 points) Draw the breadth-first search tree of G that results from carrying out a breadth-first search starting from from vertex a. In the figure you draw, show all the tree edges, and write next to each vertex both its original name and its breadth-first number.

Date due: October 30, 2012