CS381 **Homework 4** **Fall 2013**

Question 1. (30 points) Let G be a directed graph whose vertex set is $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o\}$ and whose adjacency lists representation is given below.

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

For example, the $L[c]$ list encodes the fact that vertex c is the tail of the two directed edges (c, d) and (c, h). 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 given orders for list contents in this and other questions).

1. (10 points) Draw the depth-first search tree of G that results from carrying out a depth-first search starting from vertex a. In the figure you draw, show all the tree edges as solid, the non-tree edges as dotted, and write next to each vertex both its original name and its depth-first number.

2. (10 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. (In this and all other questions of this homework, use the original names of vertices to provide your answers.)

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).

Question 2. (15 points) For the directed graph of the previous question, draw the breadth-first search tree that results from carrying out a breadth-first search starting from vertex a. In the figure you draw, show all the tree edges as solid, the non-tree edges as dotted, and write next to each vertex both its original name and its breadth-first number.

Question 3. (40 points) Let G be an undirected graph whose vertex set is $\{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o\}$ and whose adjacency lists representation is given below.

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

1. (10 points) Draw the depth-first search tree of G that results from carrying out a depth-first search starting from vertex a. In the figure you draw, show all the tree edges as solid, the non-tree edges as dotted, and write next to each vertex both its original name and its depth-first number.

2. (10 points) Recall that, in the algorithm for bridges and articulation points that we covered in class, the depth-first search is modified so as to compute, for each vertex u, a quantity $f[u]$ defined as follows (where $d[w]$ denotes the depth-first number of a vertex w):

   $f[u]$ is the smaller of $d[u]$ and of the minimum $d[w]$ such that there is a non-tree edge (v, w) where v is descendant of u.

   Write down the contents of the f array when the depth-first search of G terminates (i.e., $f[a] = 1$, $f[b] = \ldots$etc).

3. (10 points) List the bridges in the order in which they are discovered by the algorithm we covered in class. (Recall that a bridge is an edge whose removal disconnects the graph.)

4. (10 points) List the articulation points in the order in which they are discovered by the algorithm we covered in class. (Recall that an articulation point is a vertex whose removal disconnects the graph.)

Question 4. (15 points) For the undirected graph of the previous question, draw the breadth-first search tree that results from carrying out a breadth-first search starting from vertex a. In the figure you draw, show all the tree edges as solid, the non-tree edges as dotted, and write next to each vertex both its original name and its breadth-first number.

Date due: Thursday October 3, 2013