CSCI 3104-Spring 2015: Assignment #8.

Due date: Friday 4/3/2015, 3:30 PM (turn it in ECOT 724, CSCI 3104 mailbox)
Maximum Points: 70 points + 5 for legibility.

Note: This assignment must be turned in on paper, before class. Please do not email: it is very hard for us to keep track of email submissions. Further instructions are on the class page: http://csci3104.cs.colorado.edu

P1 (25 points) Consider the graph $G$ below:

Suppose we perform a depth first search on the graph $G$, as follows:

1. DFS is started at node $n_1$

2. Whenever it visits a node $n_j$, it explores the children $n_{j_1}, \ldots, n_{j_k}$ in ascending order of their indices $j_1 < j_2 < \cdots < j_k$.

(A) Complete the following sequence of nodes visited by the DFS along with the sequence of return nodes.

1. Call $n_1$.

2. Call $n_2$.

3. Call $n_4$. 

;
4. : (your answer should fill in the remaining calls and returns here)

(B) Write down the discovery and finish time for each of the nodes. Arrange your answer in a table of the following form:

<table>
<thead>
<tr>
<th>Node id</th>
<th>Discovery Time</th>
<th>Finish Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>n2</td>
<td>2</td>
<td>?</td>
</tr>
<tr>
<td>n3</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n11</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

(C) Draw the DFS tree. Identify all the cross edges, forward edges and back edges.

(D) Write down the forward and backward reachable subsets for each of the following nodes: $n_1, n_4, n_{10}$.

(E) Identify all the (maximal) strongly connected components of this graph.

**P2 (15 points)** We performed DFS on an unknown graph $G$ with 7 nodes, and obtained the following discovery and finish times:

<table>
<thead>
<tr>
<th>Node</th>
<th>Discovery</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

(A) Draw the DFS tree for the graph.

(B) Given the information provided above, for each of the possible edges given below, write down whether “the edge definitely exists”, “the edge definitely does not exist” or “the edge may/may not exist”. Give a single sentence reason for your answer.

1. (7, 1)
2. (1, 2)
3. (3, 6)
4. (6, 3)
5. (1, 6)

**P3 (15 points)** Consider a task dependency graph shown below. Vertices correspond to tasks and edge $(T_i, T_j)$ means that task $T_i$ can be performed only after $T_j$ can be performed.
1. Perform a DFS search starting at node $T_1$. Write down the list of visited nodes along with their discovery and finish times.

2. Now, continue your DFS search starting at node $T_2$. Write down the list of visited nodes along with their discovery and finish time. Note that the discovery time for $T_2$ itself will be 1+ finish time for $T_1$ in the first search.

3. Write down the topologically sorted list of nodes using the DFS finish time.

**P4 (15 points)** A mystery language has $k$ symbols in its alphabet. But all you have is a dictionary for this language and you are given that the words in the dictionary are sorted in lexicographic order (as all dictionaries are).

Based on the given list of words in the dictionary order, write an algorithm to extract a possible alphabetic order of symbols for the mystery language?

Demonstrate your solution on the following input list with alphabet #, !, %,.

```
##!*#
#%*#!
#%!!**
#!%**
#!**%
*##!**
!!***
```

**Bonus Problem (will not be graded and no credit)** You are given a list $A$

$[6, 1, 19, 3, 10, 12, 32, 17, 2, 30]$


For instance, $A[1] = 1$ and $A[8] = 2$ are adjacent. Therefore $(1, 8)$ is an adjacent pair.

Likewise, $(2, 9)$ is another adjacent pair.

Write an efficient algorithm that given the list $A$ will print all the adjacent pairs.