COMPSCI 220 S2, 2015

Assignment 3 **Due: 23rd October 2015, 8.30 pm**

1. *A graph-theoretic problem.*

   Seven students Ben (B), Don (D), Felix (F), June (J), Kim (K), Lyle (L), and Maria (M) are looking for jobs after graduation. The University Placement Office has posted open positions for an accountant (a), consultant (c), editor (e), programmer (p), reporter (r), secretary (s), and teacher (t). Each of the seven students has applied for some of these positions as follows:

   B: c, e   D: a, c, p, s, t   F: c, r   J: c, e, r
   K: a, e, p, s   L: e, r   M: p, r, s, t

   Is it possible for each student to be hired for a job for which they have applied?

   To get full marks, your answer to this question should be clear and detailed. In particular, you are asked to explain which graph-theoretic concept can be used to model the above situation, apply this concept to the situation, and explain how the resulting graph can be exploited to answer the question.

   (3 marks)

2. *Connectivity.*

   Let D be a digraph, and let G be a graph. Recall that the *underlying graph* of D is obtained by removing all directions from the arcs of D and replacing any resulting pair of parallel edges by a single edge. Moreover, an *edge-cut* of G is a set S of edges of G such that $G - S$ is disconnected, where $G - S$ denotes the graph obtained from G by deleting all edges in S.

   Now, let D be a digraph with at least two vertices. Prove that D is strongly connected if and only if for every edge-cut S of the underlying graph G of D that separates $V(G - S)$ into two sets A and B, there is an arc in D directed from a vertex in A to a vertex in B and an arc in D directed from a vertex in B to a vertex in A.

   Be as detailed and precise as possible in writing up your proof!

   (7 marks)

3. *Graph algorithms.*

   Write two algorithms to solve the following two algorithmic tasks for DAGs.

   (a) Determine the number of connected components in the underlying graph.

   (b) Find the length of the longest path starting from vertex 0.

   Please provide a separate program to solve each task.

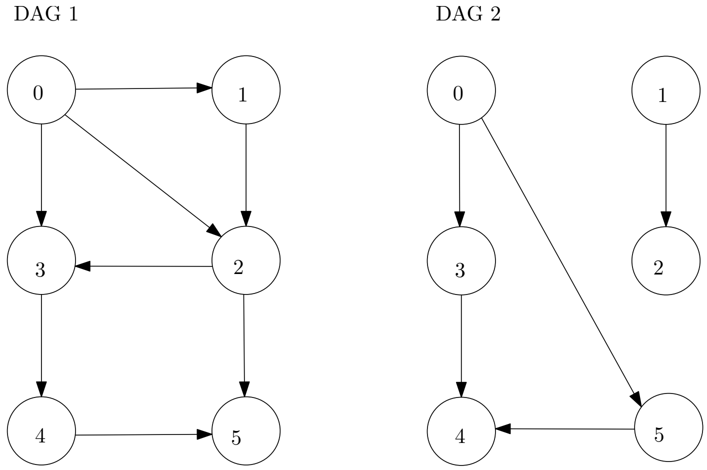
   Input format. The input for each task consists of a sequence of one or more DAGs. Each DAG D is represented by an adjacency list. The first line is an integer n that indicates the order of D. This is followed by n white space separated lists of adjacencies for vertices labeled 0 to $n - 1$. The input is terminated by a line consisting of a single zero. This line should not be processes. Each input DAG may contain up to 5000 vertices and 10000 arcs. An example input that contains two DAGs is shown next.

```
6
1 2 3
2
3 5
4
5

6
3 5
2

4

4
0
```



Output format. The output is a sequence of lines, one for each input DAG. Each line contains a single integer indicating the answer to one of the two tasks.

Submit your source code to the automated marker `www.cs.auckland.ac.nz/automated-marker`. Name them `task1.java and task2.java`, where the filename extension `java` may be replaced with extensions `cpp, py` or `cs` for C++, Python, C#/mono compilers. There will be a test case provided for each task. Note that for each task, one reads input from stdin/keyboard and prints the output to stdout/console/screen. Your programs should be efficient, i.e. they must complete with the correct answer within 10 seconds for any feasible type of input!

(10 marks)

The deadline is 8.30 pm (automarker time) on October, 23. The marks for this assignment are worth 9% of your course grade. The due date is firm in that there are no penalty options available for late submissions (except zero marks).