Computer Science 220S2C (2018)

Assignment 2 (Graphs and traversals)

Due date September 28, 2018, 11pm

1 Written submission

(60 marks out of 100)

1. *(10 marks total: 1 mark each)* Consider the digraph G defined by the adjacency list

```
0: 3 5 7
1: 2 4
2: 7
3: 2 5
4: 1
5: 6
6: 0
7: 0
```

   (a) What is the order of G?
   (b) What is the size of G?
   (c) What is the size of the underlying graph of G?
   (d) What is the distance from node 2 to node 3?
   (e) What is the eccentricity of node 3 in the underlying graph of G?
   (f) What is the diameter of the underlying graph of G?
   (g) What is the radius of the underlying graph of G?
   (h) What is the longest path in G?
   (i) What is the longest cycle in G?
   (j) What is the sub-digraph of G induced by selecting nodes $V' = \{2, 3, 7\}$?

2. *(30 marks)* Consider the digraph G with nodes $\{0, 1, 2, 3, 4, 5\}$ and adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

   (a) *(15 marks)* Perform a breadth first traversal and a depth first traversal of G. Where there is a choice of nodes to pick, pick the one with the smallest label. Show your working and present the resulting search forests as two pred arrays.

   (b) *(5 marks)* For each traversal, classify all arcs of G as tree, forward, back or cross arcs.

   (c) *(5 marks)* Explain how to determine, at the time when an arc is first explored by DFS, whether it is a cross arc or a forward arc.

   (d) *(5 marks)* Explain how to distinguish between a back and a cross arc while BFS is running on a digraph.

3. *(20 marks)* There is a proof in the textbook (Exercise 5.6.2) that every DAG has a source and a sink. Prove this result using a different argument.

Submission of written part

You should submit via Canvas, the following:

- A single PDF file containing your answers. Do the best you can with mathematical symbols. For exponents, write something like 2^n if using plain text. Use LaTeX for high quality.
- A scanned handwritten submission is acceptable and may be easier if you want to draw graphs but please make sure it is neat and the scan is easy to read.
- Any results from the textbook or lecture notes that you use must be clearly cited and all working should be shown.

2 Automarker Submission

(40 marks out of 100)

This section of the assignment requires you to implement a method to find the girth of a digraph. To get marks for this question you need to submit a program that you have written yourself to `http://www.cs.auckland.ac.nz/automated-marker`. You must

implement your method from first principles and cannot use an existing library method that finds the girth. Note this marker runs on a linux box so no Microsoft specific code should be used; read the automarker help and FAQ for more details.

Please submit source code named girth.ext where ext denotes one of {java, cpp, py, cs} to indicate a java/c++/python/c#(mono) program. An excessive number of submissions (over 10) will incur a 20% penalty for the problem if you eventually solve it.

Girth of a digraph

For a digraph G, the length of the shortest cycle is called the directed girth of the graph. If the digraph has no cycle then the girth is undefined, and we will say it has girth 0. Your task is to determine the directed girth of each input digraph.

Input Format

Input for this problem consists of a sequence of one or more digraphs taken from the keyboard (System.in). Each graph is represented by an adjacency list. The first line is an integer n indicating the order of the graph. This is followed by n white space separated lists of adjacencies for nodes labeled 0 to n - 1. The input will be terminated by a line consisting of one zero (0). This line should not be processed. Two sample input digraphs are listed below. The test cases are digraphs of order at most 30.

```
4
1 3
2 3
0

3
1 2

1
0
```

Output Format

Output will be just one integer per line sent to the console (e.g. System.out). For the above, input we would output the following two integers denoting the girth of the two graphs, using 0 to indicate the graph has no cycle.

```
3
0
```