

Computer Science 220S2C (2022)

Assignment 3 (Basic graph algorithms and traversals)

See Canvas for due date

This assignment requires you to submit programs in Python that you have written yourself to the automarker, <https://www.automarker.cs.auckland.ac.nz>. Your implementation must be from first principles and cannot use an existing library methods that might solve the problem (eg performs graph operations etc).

The automarker runs on a Linux box. Read the automarker help and FAQ for more details.

Please submit only Python source code (.py extensions only).

1. **Delete a node from a digraph** *30 Marks*

For each digraphs in a set of digraphs with order at least 3 given as adjacency lists, read in the digraph, delete the node with index $n - 3$ where n is the order of the digraph, and write the digraph back to file.

Input format: described below under the heading “Digraph input format”.

Output format: the same as the input format. Ensure that you maintain the node naming conventions.

For the example input shown below, the first digraph would have node with index 1 removed, and the second graph would have node index 0 removed, so the output would be

```
3
2
0

2

0
0
```

2. Forward and cross arcs in a DFS *30 Marks*

For a given set of digraphs, write a program that performs DFS on each digraph starting at node 0 and prints out the first forward arc and first cross arc encountered in the traversal. Use our standard convention that when there is a choice of white or grey nodes, the one with the lowest index should be chosen.

Input format: described below under the heading, “Digraph input format”.

Output format: For each input digraph, print out two lines. The first line has the first forward arc encountered, the second line has the first cross arc encountered. Write the arc (u, v) as **u,v**. If the required forward or cross arc does not exist, output an empty line in its place.

For the example input shown below, the output (including blank lines) would be

0,3

2,1

3. BFS to find distances *30 Marks*

Write a program that performs BFS on each of a given set of digraphs starting at node 0 and prints the distance to the most distant node from 0 and reports the node with the lowest index at that distance. Nodes that are not reachable from 0 have an undefined distance and should be ignored.

Input format: described below under the heading, “Digraph input format”.

Output format: For each input digraph, print out a line with the distance to the most distant node, then a space, then the lowest index of a node at that distance. Ignore nodes that are not reachable from 0.

For the example input shown below, the output would be

2 2

1 1

Digraph input format

A sequence of one or more digraphs is 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 lists are sorted. The input will be terminated by a line consisting of one zero (0). This line should not be processed. The sample input below shows two digraphs, the first has node set $\{0, 1, 2, 3\}$ and arc set $\{(0, 1), (0, 3), (1, 2), (1, 3), (2, 0)\}$, the second has node set $\{0, 1, 2\}$ and arc set $\{(0, 1), (0, 2), (2, 1)\}$.

```
4
1 3
2 3
0
```

```
3
1 2
```

```
1
0
```

Marking

The maximum number of submissions for each problem is fixed at 12.

Each problem has 3 test cases associated with it worth one third of the marks for that problem. Some of the test cases will be large. You get full marks if you pass all test cases.