

There are **three** problems listed below. To get full credit for this assignment you need to complete all of them!

If you are stuck or confused by any of the problems, feel free to post on Piazza!

Show all working; if you do not show your work the mark will be reduced. You should submit via Canvas a **SINGLE PDF** file containing the answers to the written questions. A scanned handwritten submission is acceptable if and only if it is very neatly written. If typing the assignment, do the best you can with mathematical symbols. For exponents, write something like

$2^n$

if using plain text. Use LaTeX if you really want it to look good.

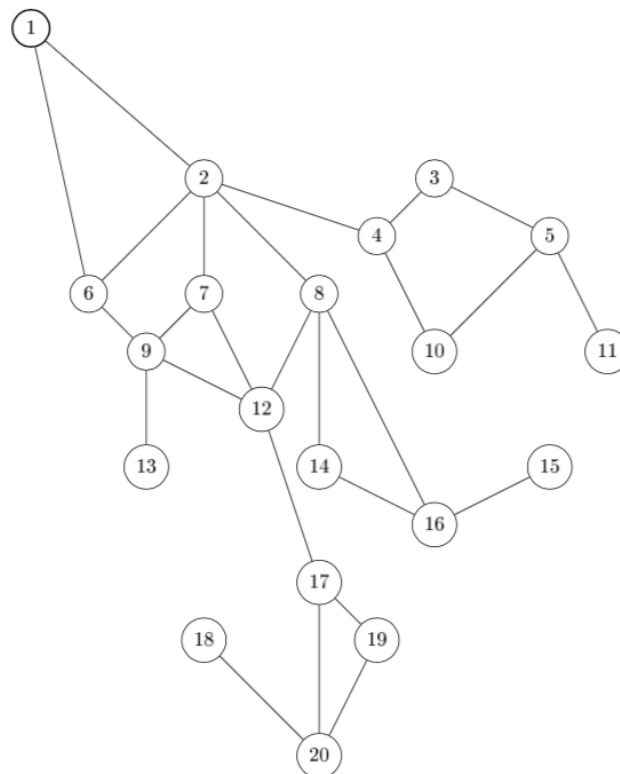
Answers to programming questions must be submitted via the automated marker system:

`automarker.cs.auckland.ac.nz`

Best of luck, and enjoy the problems!

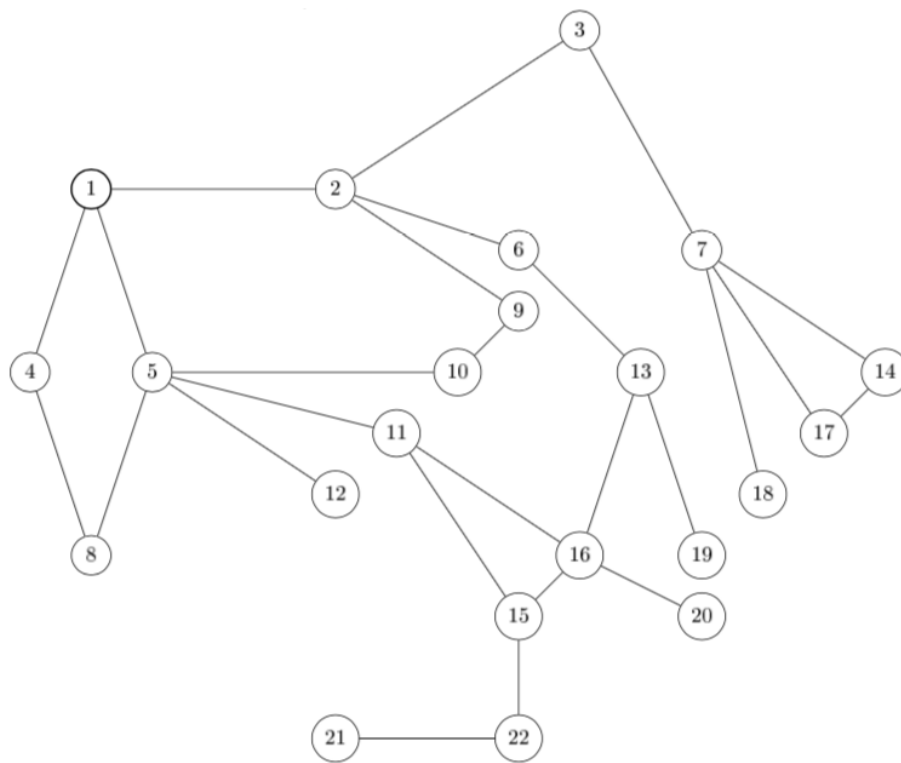
1. Apply DFS to the following graph and give the order of the vertices in which they get visited. If there are several choices, choose the vertex with the smallest number.

(1 mark)



Apply BFS to the following graph and give the order of the vertices in which they get visited. If there are several choices, choose the vertex with the smallest number.

(a) (1 mark)

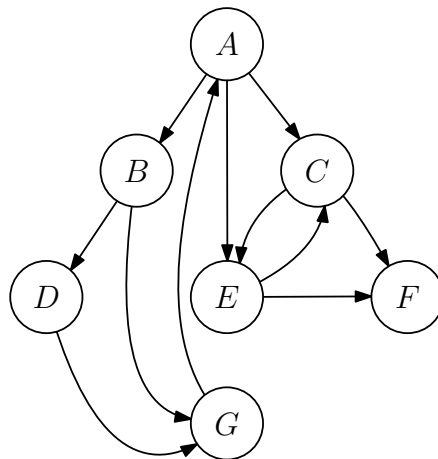


- (b) Draw two distinct graphs  $G_1$  and  $G_2$  on 7 vertices  $\{a, b, c, d, e, f, g\}$  such that (i) a DFS starting at  $a$  of  $G_1$  visits the vertices of  $G_1$  in the same order as a BFS starting at  $a$  of  $G_1$ , AND (ii) a DFS starting at  $a$  of  $G_2$  visits the vertices of  $G_2$  in the same order as a BFS starting at  $a$  of  $G_2$ .

Note. If DFS has several choices, it chooses the vertex that comes first in the alphabetical ordering. For example, if  $c$  and  $e$  can both be visited next, DFS chooses  $c$ .

(2 marks)

- (c) Consider the following digraph  $G$ .



- List all vertices in  $G$  in the order they are visited when running a BFS search on  $G$  that starts at  $A$  and picks nodes in alphabetical ordering when several choices are possible.
- Considering the order with which the BFS search in (a) has visited the nodes of  $G$ , classify each arc in  $G$  as a tree, forward, back, or cross arc.

(1 mark)

(2 marks)

## 2. AVL trees

(6 marks)

For a given adjacency matrix of a rooted tree you need to decide whether this tree can be labeled to become an avl tree. Vertices of the tree do not have keys but numbered from 0 to  $n - 1$  in order to write the adjacency matrix.

**INPUT format:** The input consists of blocks. Blocks are written one after another without empty lines between them. Every block starts with a new line and consists of several lines:

- the first line contains a positive integer number  $n$  which corresponds to the number of nodes in a tree and the root index;
- the next  $n$  lines correspond to the adjacency matrix of this tree.

**OUTPUT format:** The answer for every block should consist of the height of the tree and True/False (True if the tree is an avl tree). Answers for different blocks should be located in different lines.

### 3. Page ranking

(7 marks)

You are asked to create a simplified version of the Google page ranking algorithm. You are given a map of links between  $n$  websites which is represented by a digraph. Websites are nodes marked from 0 to  $n - 1$ . An arc from  $i$  to  $j$  means that website  $i$  has a link to website  $j$ . Suppose you are searching for something and the result of your search is these  $n$  websites. Usually the search engine outputs these websites in order of popularity. In this problem, let the popularity index of a website be the number of websites which have a link to this website. So the first website in the list is the one with the largest popularity index. The one after it is the one with the second largest popularity index, etc. If there are several websites with identical popularity index, then they are ordered according to their numbers in the ascending order. You need to output the number of the  $k$ th website in this list. Note that  $k$  is between 1 and  $n$  inclusive.

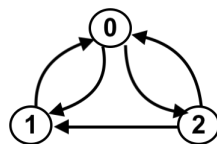
**INPUT format:** The input consists of blocks. Blocks are written one after another without empty lines between them. Every block starts from a new line and consists of several lines:

- the first line contains a positive integer number  $n$  which corresponds to the number of websites in this block and number  $k$ ,  $1 \leq k \leq n$ ;
- the next  $n$  lines correspond to the adjacency list of the digraph.

For example, the block:

3	1
1	2
0	
0	1

corresponds to the linking map graph



You need to find the first most popular website, which is 0.

**OUTPUT format:** For every block you need to output the  $k$ th most popular website. Answers for different blocks should be located in different lines.

## Questions involving programming

- Supported languages are Java, C, C++, PyPy, Python, Go, C#, Rust, F#, Javascript.
- Your answer to each question should be a single file (containing all nonstandard classes you use).

- A sample input and output file for each question will be available from Canvas. The markers may check the output with a text comparison program, so it must be in EXACTLY the right format.
- The input is taken from stdin.
- You have an **unlimited** number of submissions.
- After the 10th submission there is a penalty of 25%.
- After the 20th submission there is a penalty of 50%. That is if you got the result correct only on your 21st try, then you get 50% of the mark. This adjustment will be made by instructors after the assignment due date.
- If you were not able to pass the automarker's test you still have a chance to get up to 25% for detailed explanation of your algorithm. Markers will check your code for comments only if you have not passed the automarker.