CSE431/531 (Fall 2022) Programming Assignment 1

Due date: Monday, 3/28/2022

You are given an (undirected) graph $G = (V, E)$, which is not necessarily connected. You need to use breadth-first-search or depth-first-search to output the number of connected components of G, and the size of the largest component, where the size of a component is defined as the number of vertices in it. Notice that an isolated vertex is counted as a connected component of size 1.

You can use C++, Java or Python to implement the algorithm.

- **Input**: You need to read the input graph from the terminal (i.e, the standard input). In the first line of the input, there are two positive integers n and m. n is the number of vertices in the graph and m is the number of edges in the graph. The vertices are indexed from 1 to n. You can assume that $1 \leq n \leq 10000$ and $1 \leq m \leq 100000$. In the next m lines, each line contains 2 integers: u and v, with $1 \leq u < v \leq n$. This indicates that there is an edge (u, v) in the graph G. It is guaranteed that the graph has no self-loops and parallel edges.
- **Output**: You need to print your result to the terminal (i.e, the standard output). The output contains two lines, each containing a single integer. The integer on the first line is the number of connected components G has, and the one on the second line is the size of the largest connected component.

Input #1:	**Output #1**:	**Input #2**:	**Output #2**:
4 3 1 2 2 3 1 3	2 3	8 7 1 2 3 4 5 6 1 7 7 8 2 7 2 8	3 4

For input 1, the graph has 2 connected components, with vertex-sets $\{1, 2, 3\}$ and $\{4\}$ respectively. For input 2, the graph has 3 connected components, with vertex-sets $\{1, 2, 7, 8\}, \{3, 4\}$ and $\{5, 6\}$ respectively.