Components in a graph ★

149 more points to get your next star!

Rank: 487520 | Points: 326/475

Problem Solving ★★★

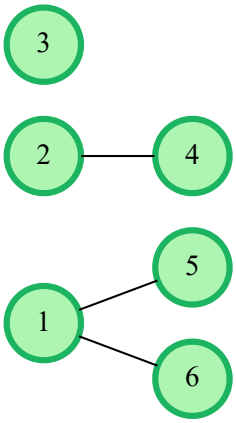
Problem Submissions Leaderboard Editorial

There are $2 \times N$ nodes in an undirected graph, and a number of edges connecting some nodes. In each edge, the first value will be between 1 and N, inclusive. The second node will be between $N+1$ and $2 \times N$, inclusive. Given a list of edges, determine the size of the smallest and largest connected components that have 2 or more nodes. A node can have any number of connections. The highest node value will always be connected to at least 1 other node.

Note Single nodes should not be considered in the answer.

Example

$bg = [[1, 5], [1, 6], [2, 4]]$



The smaller component contains 2 nodes and the larger contains 3. Return the array $[2, 3]$.

Function Description

Complete the connectedComponents function in the editor below.

connectedComponents has the following parameter(s):

- int bg[n][2]: a 2-d array of integers that represent node ends of graph edges

Returns

- int[2]: an array with 2 integers, the smallest and largest component sizes

Input Format

The first line contains an integer n, the size of bg.

Each of the next n lines contain two space-separated integers, $bg[i][0]$ and $bg[i][1]$.

Constraints

- $1 \leq numberofnodesN \leq 15000$
- $1 \leq bg[i][0] \leq N$
- $N+1 \leq bg[i][1] \leq 2N$

Sample Input

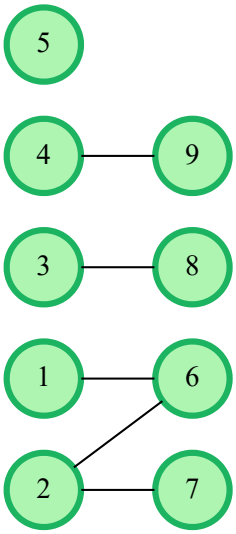
```
STDIN   Function
-----   --------

5       bg[] size n = 5
```

```
1 6    bg = [[1, 6],[2, 7], [3, 8], [4,9], [2, 6]]
2 7
3 8
4 9
2 6
```

Sample Output

```
2 4
```

Explanation



Since the component with node 5 contains only one node, it is not considered.

The number of vertices in the smallest connected component in the graph is 2 based on either $(3, 8)$ or $(4, 9)$.

The number of vertices in the largest connected component in the graph is 4 i.e. $1 - 2 - 6 - 7$.

Change Theme Language Java 8

```
4
5       public static void main(String[] args) {
6
7           Scanner sc = new Scanner(System.in);
8           int n = sc.nextInt();
9           ArrayList<Integer> a[] = new ArrayList[2*n];
10
11          for(int i=0;i<2*n;i++)
12              a[i] = new ArrayList<Integer>();
13          boolean flag[] = new boolean[n];
14          for(int i=0;i<n;i++)
15              {
16              int x = sc.nextInt();
17              int y = sc.nextInt();
18              flag[x-1]=true;
19              a[x-1].add(y-1);
20              a[y-1].add(x-1);
21          }

22          int max = 0;
```

```
24          int min = Integer.MAX_VALUE;
25
26          for(int i =0;i<n;i++)
27              {
28              int ans=1;
29              if(!flag[i])
30                  continue;
31              boolean an[] = new boolean[2*n];
```

Line: 57 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
5
1 6
2 7
3 8
4 9
2 6
```

Your Output (stdout)

```
2 4
```

Expected Output Download

```
2 4
```