CS2705 : Programming and Data Structures

Assignment 3

November 3, 2022

Submission date: Nov 5, 05:00 PM Max. Marks : 8

Instructions

- The assignment is graded. All questions are compulsory and have to be solved individually.
- You are required to submit the code on repl.it by following the given instructions.
- You are expected to write code completely on your own. Use of unfair means found will be penalized and reported to appropriate higher level committees.

1. (3 points) In a palace, there are n rooms. Some pairs of rooms are connected by paths but each path is a one-way. Any room has at most $n-1$ entry and at most $n-1$ exit gates. A room is a *secure* room if either the room has no exit gate or all exit gates lead to a room which is secure.

   The rooms are numbered from 0 and $n-1$. Print the number of secure rooms in ascending ordering of their numbers. You need to implement the algorithm with running time $O(m+n)$ where m denotes the number of paths.

   Input format. The first line contains the value of n and the number of paths (m) separated by a white-space. For the next m lines, each line contains two numbers, say a b, separated by a white-space to indicate a one-way path from room number a to room number b.

   Output format. The numbers of secure room in ascending order separated by white-spaces.

   Example.

   Input:

   7 7

   0 1

   0 2

   1 2

   1 3

   2 5

   3 0

   4 5

   Output:

2 4 5 6

Explanation:

Room number 5 and 6 are secure rooms since there is no exit gate for these rooms. Room number 2 and 4 are secure since each has a single exit gate that leads to the secure room (5). Room number 1 has two exit gates, only one of which leads to the secure room (2), hence 1 is not a secure room.

2. (3 points) On an island, there are n cities. Some pairs of cities are connected by paths but each path is a one-way. Each city has at most one path going to another city that connects it to the other city.

   Print the maximum number of cities that are reachable from each other. Implement an algorithm that takes $O(n)$ time.

   Input format. The first line contains the value of n and the number of paths (m) separated by a white-space. For the next m lines, each line contains two numbers, say a b, separated by a white-space to indicate a one-way path from city a to room b.

   Output format. The maximum number of cities that are reachable from each other.

   Example.

   Input:

   5 5

   0 3

   1 3

   2 4

   3 2

   4 3

   Output:

   3

   Explanation: Three cities $2, 3, 4$ are reachable from each other and there is no larger set.

3. (2 points) Consider a set of people on a social media platform. Two people either mutually know each other or they don't. Let $k \geq 2$. Two individuals A and B are *k-strangers* if A and B do not mutually know each other and for every t, $1 \leq t < k$, there do not exist t people $C_1, C_2, \ldots, C_t$ such that A and C_1 mutually know each other, C_1 and C_2 mutually know each other, and so on, and C_t and B mutually know each other.

   Given a set of n people and a value of k, find out the number of pairs of people who are mutually k-strangers. Your algorithm should run in time $O(n(n+m))$ where m is the number of mutually known pairs.

   Input format. The first line contains n, k and the number of pairs of people who mutually know each other (p), each separated by a white-space. For next p lines, each line contains two numbers a and b, separated by a white-space indicating that individual a and b mutually know each other. Assume that the individuals are numbered from 0 to $n-1$.

   Output format. The number of pairs who are k-strangers.

   Example.

   Input:

   6 2 7

   0 1

   0 5

1 2

1 5

2 3

3 4

4 5

Output:

1

Explanation:

For every pair (p, q) except for the pair $(0, 3)$, either p and q mutually know each other or there exists a person w such that p and w mutually know each other and w and q mutually know each other.

Example.

Input:

7 3 7

0 1

1 2

2 3

3 4

4 5

5 6

6 0

Output:

0

Explanation:

For every pair (p, q), either p and q mutually know each other or there exists t people (where $1 \leq t \leq 2$) such that each consecutive pair in sequence $p, C_1, \ldots, C_2, q$ mutually know each other.

4. (5 points) **Make-up question** Consider n individuals who are from a research community. A set S of k individuals denotes a group of *specialists*. An individual b is considered to be an *expert* if b is a specialist or b has co-authored a research paper with at least one expert. Assume that each paper is co-authored by exactly two individuals. When a new paper is written, possibly more people become experts.

   As research makes progress, a sequence of c operations are performed. Each operation is one of the two types: new-paper and is-expert. They are defined as follows –

   - new-paper is an **update** defined for two individuals u and v and it indicates that a new paper is co-authored by u and v.
   - is-expert is a **query** that returns "yes" if an individual u is an expert, "no" otherwise.

   The task is to process this sequence of operations and answer the queries in the order asked.

   Assume that people are numbered starting 0. If n denotes the number of individuals and c denotes the number of operations (updates and queries together) then your algorithm should run in time $O(c + n \log n)$

   Input format. The first line contains n, k and c, each separated by a white-space. The second line contains k numbers denoting the vertices in set S, each separated by a white-space. The sequence of c

operations begin at the third line. In each line the name of the operation and its arguments are given, each separated by a white-space.

Output format. A sequence of “yes” and “no” which denote the answers to the queries is-expert in the same order as they appear in the input, one per line.

Example.

Input:

```
10 3 11
0 5 7
is-expert 4
new-paper 0 2
new-paper 0 6
is-expert 8
new-paper 6 4
new-paper 6 7
is-expert 4
new-paper 9 8
is-expert 9
new-paper 0 8
is-expert 9
```

Output:

```
No
No
Yes
No
Yes
```

Explanation: At the time of the first two queries, 4 and 8 are neither specialists nor have co-authored a paper with an expert. At the time of the third query, 4 is an expert because 4 and 6 have co-authored a paper, and 6 is an expert. At the time of the forth query, 9 is not an expert. But at the time of the fifth query, 9 becomes expert since 9 and 8 have co-authored a paper (earlier) and (recently) 8 became an expert.