



Princeton Computer Science Contest – Spring 2023

Problem 5: The Missing Electron (25 points) [Codeforces Submission]

By Pedro Paredes

Albert has been studying atoms and molecules lately, but he seems to have lost his favorite electron in one of the atoms! He created a laser device to help him find it, but he is not quite sure how to use effectively.

Each molecule can be modeled as a simple connected undirected graph G with n vertices (atoms) and m edges (chemical bonds). Each vertex is uniquely numbered with an integer from 1 to n . The graph G is known to Albert, but the vertex (atom) containing the hidden electron is not. The device Albert built works in the following way: given a vertex v from G , Albert can shine a laser into v 's atom and find out which of v 's neighbors is closest to the vertex containing the missing electron. If there are multiple vertices at the same distance to the vertex containing the missing electron then the laser returns an arbitrary one among them. If the laser is applied to the vertex containing the missing electron the laser will flash indicating the missing electron was found, which means Albert can finally stop his search.

Albert wants you to help him finding the missing electron using the device he built. Unfortunately, the device is very fragile, and will stop working after 40 uses. Thankfully, the structure of G is not arbitrary, the molecule either follows a cyclic, tree-like or mesh-like. Each one of these is described in the following sections.

This is an interactive problem where you will write a program that interacts with an interactive grader program to find the missing electron. Instructions on how to use the grader are specified below.

Additionally, this is a multitest case problem, which means that for a single G your program is supposed to simulate multiple instances of finding the missing electron. This shall be specified in the Input and Interaction sections.

Princeton Computer Science Contest – Spring 2023





Princeton Computer Science Contest – Spring 2023

Input

The first line contains two integers separated by spaces, n ($3 \leq n \leq 5,000$) and m ($n-1 \leq m \leq n(n-1)/2$) — the number of vertices (atoms) and edges (chemical bounds) in the graph (molecule).

The next m lines contain two integers separated by spaces, u and v ($1 \leq u, v \leq n$) — the two endpoints of each edge in G .

The next line contains one integer s ($1 \leq s \leq 50$) — the number of simulations your program should support. This means your program should find s missing electrons on the same graph G , each one independently. Between each independent simulation, the number of times you can use the laser resets, so you can use it 40 times per simulation.

It is guaranteed that G is connected and simple, so there are no repeated pairs of endpoints u and v , and additionally u and v are always distinct.

Interaction

For each simulation you will alternate with the interactive grader by applying the laser at most 40 times.

- When it is your turn, you should print a single line containing an integer p ($1 \leq p \leq n$) indicating you want to apply the laser device to vertex p .
- When it is the grader's turn, you should read an integer, which will be the neighbor of p that is closest to the vertex containing the missing electron.

Once you apply the laser containing the missing electron, the interactive grader will output the same vertex you picked, indicating you found the missing electron. At this point the current simulation will end and the next will start (or the grader will terminate, in case this was the last simulation). If at any point one of your interactions is malformed (for example, if you print something other than an integer between 1 and n) or if you use the laser more than 40 times, the interactor terminates immediately and your program receives the verdict **WRONG-ANSWER**.

Princeton Computer Science Contest – Spring 2023





Princeton Computer Science Contest – Spring 2023

When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the grader. In order to avoid such situation you have to use special flush operation each time you output some data. To flush the output, use:

- `fflush(stdout)` in C;
- `fflush(stdout)`, `cout << flush` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `sys.stdout.flush()` in Python;

Part I - Cyclic Molecules (5 points)

For this part you can assume that the graph G is a cycle graph, meaning it forms a connected graph where each vertex has exactly two neighbors.

Examples

Input:

```
5 5
2 4
4 5
5 3
3 1
2 1
2
4
1
1
4
```

Princeton Computer Science Contest – Spring 2023



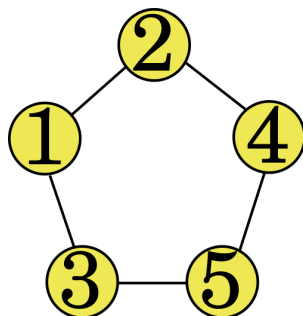


Princeton Computer Science Contest – Spring 2023

Output:

```
2
1
2
4
```

In this test case the graph has 5 vertices and 5 edges and vertex 2 is adjacent to 4, vertex 4 is adjacent to 5, vertex 5 is adjacent to 3, ... The image below represents this test case.



For this example $s = 2$, which means there are two simulation (i.e. you have to find the location of 2 electrons). The first electron is hidden in vertex 1 and the second one in vertex 4 (this information isn't given in the input, it's what you have to determine). The laser is first applied to vertex 4, and vertex 2 is the result, which means that 2 is the neighbor of 4 that is closest to the vertex containing the electron (which is 1). Then it is applied to vertex 1, which is the vertex containing the electron, so the result is 1.

Part II - Tree-like Molecules (10 points)

For this part you can assume that the graph G forms a tree, meaning it doesn't contain any cycles.

Princeton Computer Science Contest – Spring 2023



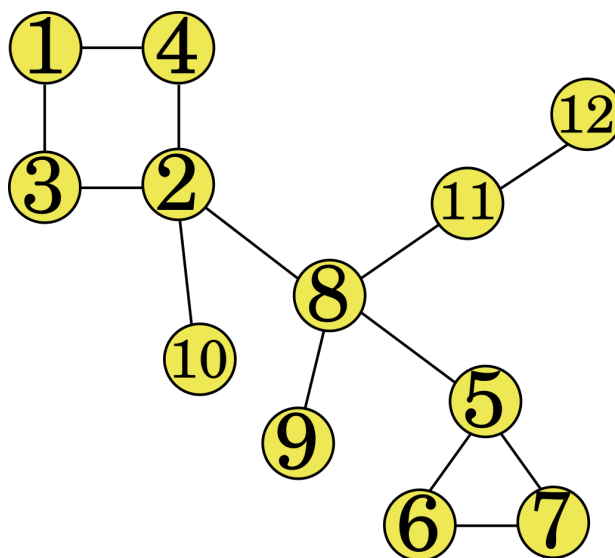


Princeton Computer Science Contest – Spring 2023

Part III - Mesh Molecules (10 points)

For this part you can assume that the graph G has a “mesh form”, which means that *all* cycles in the graph are pairwise disjoint. A graph cycle is a collection of distinct vertices v_1, \dots, v_k , such that v_1 is connected to v_2 , v_2 is connected to v_3 , \dots , and v_n is connected to v_1 . Two cycles are disjoint if they have no two vertices in common.

Here is an example of one such graph:



Princeton Computer Science Contest – Spring 2023

