IOITC 2020

Hidden Vertex

You are given a rooted tree T with n nodes numbered $1, 2, \ldots, n$. The tree is rooted at node 1. The distance between two nodes is defined as the number of edges on the unique simple path between them. There is a hidden node in the tree (say x), which you have to find. You can ask queries. In a query, you can give the judge a non empty subset S of the nodes, and in return get the sum of distances of x to the nearest and the furthest nodes from x in S.

You have to write a function:

```
int findHiddenVertex(int n, vector<int> parents)
```

where you are given a vector **parents** of length $n-1$. For each i from 0 to $n-2$, parents[i] denotes the parent of the node $i+2$. Note that it is NOT guaranteed for the parent of node i to be lesser than i. You have to return the hidden vertex in this function.

You can make calls to the function:

```
int query(vector<int> V)
```

where V is a subset of nodes, and in return get the sum of the smallest and the largest distances from x to a node in V. V is not allowed to have duplicate elements.

IMPORTANT:

- The grader is NOT adaptive. This means that, in each testcase, the hidden vertex is fixed and won't change according to the queries you ask.
- Do not print anything to stdout or stderr from within the function while submitting.
- The use of any kind of **randomness** is prohibited. Any solution seen using randomness (whether it provably works or not) will be disqualified. This includes writing your own pseudorandom generators instead of using the rand() function.

You can compile locally with

```
g++ dummy_grader.cpp dummy_solution.cpp -o grader
```

Then you can run ./grader and give input of the form as given in the sample input:

- The first line has the number of testcases.
- The first line of each testcase has the values of n and the hidden vertex.
- The second line of each testcase has the parents of the vertices $2, 3, \ldots n$ in that order.

Test data

Your code will be run against multiple testcases. In each testcase, $n \geq 2$ and the sum of n in all testcases doesn't exceed 1024.

Scoring

If for any testcase in any testfile you return an incorrect vertex, you get a score of 0. Else, in some testfile, let N_i be the value of n in the i-th testcase and Q_i be the number of queries asked by you, then:

- If $Q_i \leq \lceil \log N_i \rceil$ for all valid i, you get a score of 100 in that testfile.
- Else if $Q_i \leq \lceil \log N_i \rceil + 1$ for all valid i, you get a score of 52 in that testfile.
- Else if $Q_i \leq \lceil \log N_i \rceil + 2$ for all valid i, you get a score of 26 in that testfile.
- Else if $Q_i \leq N_i$ for all valid i, you get a score of 7 in that testfile.
- Else you get a score of 0 in that testfile.

The score of a submission would be equal to minimum of the scores in all the testfiles.

Limits

Time: 1 second

Memory: 256 MB