IOITC 2021

Edge Permutation Retrieval

There is a hidden undirected connected graph G with n nodes and m edges numbered $0, 1, \ldots m-1$, and a hidden permutation $p_0, p_1, \ldots p_{m-1}$ of edges. The graph doesn't contain any self loops or multiple edges.

You only know the value of m. You can ask queries. In one query, you give the judge a vector W of size m consisting of integers in the range $[1, 10^9]$, and the judge returns the set of edges in a minimum spanning tree of graph G if for each each $i = 0, 1, \ldots m-1$, the edge numbered p_i had a weight W_i. To make a query, you can call the function:

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

which returns a vector of size $n-1$ containing the indices of edges of a minimum spanning tree. Please note that if there are multiple minimum spanning trees, the judge can return any one of them.

You need to find the hidden permutation p or claim that it is impossible to find this permutation no matter how many queries one can ask. Implement the function:

```
vector<int> findHiddenPermutation(int m)
```

that makes queries and returns the hidden permutation p in the end. If it is impossible to find the hidden permutation, the function must return an empty vector instead.

Test Data and Scoring

Each subtask has multiple testfiles. In each testfile, your code will be run against multiple testcases. For each testcase, $m \geq 1$ and the sum of m over all testcases of a testfile doesn't exceed 1000.

The score of a subtask equals the minimum score over all its testfiles.

The score for a testfile is calculated as follows: Let m_i be the value of m, q_i be the number of queries asked by you and D_i be the maximum weight of an edge used by you in a query in the i-th testcase. If, for some i, your function returns an incorrect vector or $q_i > \max(2m_i, m_i + 5)$, you get a score of 0. Otherwise,

Subtask 1 (5 Points): The graph G is a tree. You get 5 points if $q_i \leq m_i$ for all valid i

Subtask 2 (16 Points): The graph G is a cycle. You get 16 points if $q_i \leq m_i$ for all valid i.

Subtask 3 (Total 79 points): There are no additional constraints on the graph.

- If $\max(q_i, D_i) \leq m_i$ for each valid i, you get a score of 79 points.
- Else if $q_i \leq m_i$ for each valid i, you get a score of 71
- Else if $q_i \leq m_i + 5$ for each valid i, you get a score of 59
- Else if $q_i \leq 2m_i$ for each valid i, you get a score of 18
- Else you get a score of 0.

Local testing

You are provided with a dummy grader with the name `dummy_grader.cpp`. **This grader works only if it is possible to find the hidden permutation**. You should compile your solution (assumed to be in the file `solution.cpp`) as:

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

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

- The first line contains T, the number of testcases.
- The first line of each testcase contains n and m, the number of nodes and the number of edges in the graph.
- The second line contains the expected permutation $p_0, p_1, \ldots, p_{m-1}$.
- Each of the next m lines contains an edge of the graph. The nodes are numbered $0, 1, \ldots, n-1$.

Do NOT read anything from stdin or write something to stdout/stderr.

Limits

Time: 2 seconds

Memory: 256 MB