



# Virtual Tree

The judge has a tree  $T$  with  $N$  nodes, indexed from 1 to  $N$ . You need to guess the structure of the tree by asking the judge some questions of the following type:

- Give the judge a set of nodes  $S$  and another node  $X$ . Judge will tell you whether the minimum sub-tree of  $T$  that contains all the nodes in  $S$  contains  $X$  or not.

You can ask this question at most 20000 times.

**Note:** A sub-tree of  $T$  is a connected subgraph of  $T$  that is also a tree. And the minimality is to be considered in terms of the number of nodes.

## Implementation Details

Implement the following function:

```
std::vector<std::pair<int, int>> guess_tree(int N)
```

where  $N$  is the number of nodes in the tree, and this function should return a `std::vector` of size  $N - 1$  containing the edges of the tree. Edges can be ordered or directed in any possible way.

Include the header `vt.h`. Then you can call the function:

```
bool ask(std::vector<int> S, int X)
```

which will return `true` if the minimum sub-graph of  $T$  that contains all nodes in  $S$  also contains  $X$ , `false` otherwise. Keep in mind that this function runs in  $O(|S|)$  time.

## Constraints

- $1 \leq N \leq 1000$

## Subtasks

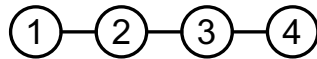
1. (8 points)  $N \leq 100$  and every node of  $T$  has degree at most 2. In other words,  $T$  is a line.
2. (5 points) Every node of  $T$  has degree at most 2. In other words,  $T$  is a line.
3. (11 points)  $N \leq 100$  and if  $T$  is rooted at node 1, each node will have an index higher than its parent's index.

4. (22 points) If  $T$  is rooted at node 1, each node will have an index higher than its parent's index.
5. (23 points)  $N \leq 100$
6. (31 points) No further constraints.

## Examples

### Example 1

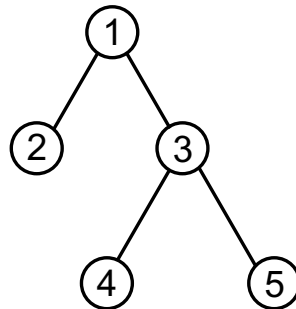
Consider the following tree:



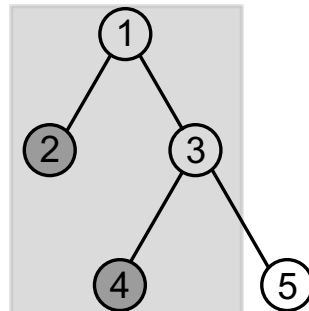
Then an acceptable return value of `guess_tree(4)` is  $\{\{1, 2\}, \{2, 3\}, \{3, 4\}\}$ .

### Example 2

Consider the following tree:



Consider  $S = \{2, 4\}$ . Then the minimum sub-tree that contains all nodes in  $S$  is highlighted bellow:



Hence, `ask({2, 4}, 3)` is true, while `ask({2, 4}, 5)` is false.

An acceptable return value of `guess_tree(5)` is  $\{\{3, 5\}, \{1, 3\}, \{2, 1\}, \{4, 3\}\}$ .

## Sample Grader

From the task page, you can download a sample grader. It consists of these files:

- `vt.h`: The shared header file.
- `vt.cpp`: The file where you should write your solution.
- `grader.cpp`: This is a grader which will simulate the judge. Your code in `vt.cpp` needs to be

compiled together with this file.

- `compile.bat`: Run this file to compile `grader.cpp` and `vt.cpp` together. Make sure you have `g++` in system's `PATH` variable. It will produce an executable `vt` in which you can give input to the sample grader.

The sample grader reads the input in the following format:

- line 1:  $n$
- line  $1 + i$  ( $1 \leq i < n$ ):  $u \ v$ , indicating an edge between nodes  $u$  and  $v$  in  $T$ .

The sample grader writes the outcome (error messages if any, query count etc.) to `stderr`. Note that the grader used in the judging server is different than the one provided here.