

# Yet Another Contest 8 P5 - Hidden Tree

**Time limit:** 2.0s    **Memory limit:** 512M

Nils has a hidden tree, which he uses to challenge Josh and Mike! The tree has  $N$  nodes and  $N - 1$  bidirectional edges, with the  $i$ -th edge connecting nodes  $u_i$  and  $v_i$ .

First, Nils will show Josh the tree. Then, Josh will create a sequence of  $N$  integers  $a_1, a_2, \dots, a_N$ , satisfying  $1 \leq a_i \leq 10^4$ .

Next, Mike will be asked  $Q$  queries about the tree. Note that Mike does not know Nils' tree or Josh's sequence, or even the value of  $N$ .

In the  $i$ -th query, Nils will select two non-adjacent nodes  $X_i$  and  $Y_i$ . Mike will be told the values of  $a_{X_i}$  and  $a_{Y_i}$ , and is challenged to respond with any integer  $Z_i$  such that node  $Z_i$  lies on the unique simple path between nodes  $X_i$  and  $Y_i$  on the tree, with  $Z_i \neq X_i$  and  $Z_i \neq Y_i$ .

Josh and Mike are working as a team, so they should decide on a shared strategy such that Josh's choice of sequence allows Mike to correctly answer his queries. When Josh is able to choose large integers, more information can be passed to Mike. Thus, Josh is challenged to minimize the maximum value in  $a_1, a_2, \dots, a_N$ , whilst still communicating enough information for Mike to answer all queries correctly.

Can you help Josh and Mike devise an optimal strategy?

## Constraints

$$3 \leq N \leq 100$$

$$1 \leq Q \leq 10^4$$

$$1 \leq u_i, v_i \leq N, u_i \neq v_i$$

$$1 \leq X_i, Y_i \leq N, X_i \neq Y_i. \text{ No edge will exist between nodes } X_i \text{ and } Y_i.$$

The edges form a tree.

## Scoring

If any query is answered incorrectly, the program will score 0 points.

Otherwise, let  $C$  be the maximum value in the sequence  $a_1, a_2, \dots, a_N$  made by your program in any test case. Let  $D = \lceil \frac{C}{100} \rceil$ . Your program will receive the following score:

$D$	Score
$1 \leq D \leq 20$	100
$20 < D \leq 100$	$\lfloor 75 - 25 \times \log_2(\frac{D}{25}) \rfloor$
$D > 100$	0

The score for the problem is equal to the minimum score on any test case.

## Grading Environment

---

Your program will be executed in two separate processes. The first process will play the role of Josh, whilst the second process will play the role of Mike. This is to ensure that no global variables are shared between the processes.

Note that the queries for the second process are based off the output of the first process. Additionally, the correctness of the second process will be judged based on the tree given as an input to the first process.

Additionally, an interactive judge is used. This means that each query must be answered before the next one is given. The interactor is not adaptive in the sense that the pairs of nodes  $X_i$  and  $Y_i$  are predetermined.

Finally, the sum of the execution times across the two processes will be compared against the time limit. The time taken by the interactor does not count towards the time limit. The memory used by each of the processes will separately be compared against the memory limit.

## Interaction

---

The first line contains a single integer, either 1 or 2.

If the integer on the first line is 1, then your program should play the role of Josh. The second line will contain a single integer,  $N$ . The  $i$ -th of the following  $N - 1$  lines will contain two space-separated integers,  $u_i$  and  $v_i$ . You should then output a single line of  $N$  space-separated integers,  $a_1, a_2, \dots, a_N$ .

If the integer on the first line is 2, then your program should play the role of Mike. The second line will contain a single integer,  $Q$ . Then, you should answer  $Q$  queries. For the  $i$ -th query, you should read in a line containing two space-separated integers,  $a_{X_i}$  and  $a_{Y_i}$ . You should then output the answer to the  $i$ -th query on a separate line. If at any point your answer is formatted incorrectly or is incorrect, the interactor will respond with `-1 -1` on a single line. After receiving this feedback, you should terminate your program to receive a Wrong Answer verdict; otherwise, your program will receive an arbitrary verdict.

**Please note that you may need to flush `stdout` after outputting your chosen array and after answering each query, or interaction may halt. In C++, this can be done with `fflush(stdout)` or `cout << flush` (depending on whether you use `printf` or `cout`). In Java, this can be done with `System.out.flush()`. In Python, you can use `sys.stdout.flush()`.**

## Sample Interaction 1

---

```
1
5
1 2
2 3
2 4
4 5
>>> 10 13 11 15 12
```

## Sample Interaction 2

---

```
2
3
10 11
>>> 2
10 12
>>> 4
11 12
>>> 4
```

In the sample interactions above, `>>>` denotes your output. Do not print this out.