Telepathy

Aitana and Bruno are visiting a national park in Bolivia. In the national park, there are N sites, and there are $N - 1$ roads connecting two sites. It is possible to move from any site to any site by passing through some roads.

When they were walking in the national park, they got separated from each other. From now on they must meet again by reaching the same site at the same time. However, being deep in the Amazon rainforest, they cannot communicate with each other. The only thing they can rely on is their own map, which depicts the road structure of the national park. Each of them wrote labels $0, 1, \dots, N - 1$ for each site in his/her map. **However, the labeling of Aitana and Bruno may be different.**

Aitana and Bruno now start moving to meet again. For each turn, they simultaneously perform either of the following actions: move to the site that is directly connected by road to the current site, or stay at the current site.

Write a program that implements a strategy to make Aitana and Bruno meet again. In this problem, a submission receives the full score if they meet again within $6d$ turns, where d is the minimum number of roads to pass to move from Aitana's current site to Bruno's current site. **Note that, when they come to the same place in the middle of the road, it is not considered that they meet again.**

In this problem, one must solve for Q scenarios in a single run of the program.

Problem Details

In this section, we formally explain the problem. Each site of the national park is assigned an ID from 0 to $N - 1$, and the j-th road ($0 \le j \le N - 2$) connects the site with ID u_j and ID v_j. For the site with ID i ($0 \le i \le N - 1$), label p_i is written on Aitana's map, and label q_i is written on Bruno's map. Here, $(p_0, p_1, \dots, p_{N-1})$ and $(q_0, q_1, \dots, q_{N-1})$ are permutations of $(0, 1, \dots, N - 1)$.

Aitana knows that, for each $j = 0, 1, \dots, N - 2$, there is a road connecting the sites labeled A_j and B_j, and Aitana is currently at the site labeled S. The "label" here is based on the Aitana's map. Hence, the j-th road ($0 \le j \le N - 2$) connects the sites labeled p_{u_j} and label p_{v_j}, and $S = p_s$ holds where s is the ID of Aitana's current site. However, the roads may not be given in the order, and the two sites that each road connects may not be given in the order of u_j, v_j. Similarly, Bruno knows that, for each $j = 0, 1, \dots, N - 2$, there is a road connecting the sites labeled C_j and D_j, and Bruno is currently at the site labeled T. The "label" here is based on the Bruno's map. Especially, $T = q_t$ holds where t is the ID of Bruno's current site.

Based on the information above, Aitana and Bruno decide their movement of the next $10N$ turns. In other words, Aitana decides the sequence of labels $x_0, x_1, \dots, x_{10N}$, and Bruno decides the sequence of labels $y_0, y_1, \dots, y_{10N}$, which represents his/her movement, independently. They must satisfy the following conditions:

- $x_0 = S$, and for each k ($1 \leq k \leq 10N$), the sites labeled x_{k-1} and x_k in Aitana's map are either the same site or directly connected by a road.
- $y_0 = T$, and for each k ($1 \leq k \leq 10N$), the sites labeled y_{k-1} and y_k in Bruno's map are either the same site or directly connected by a road.

The turn number k^* when Aitana and Bruno meet again is the minimum k such that label x_k (in Aitana's map) and label y_k (in Bruno's map) represent the same site. A submission receives the full score if $k^* \leq 6d$.

Implementation Details

You must submit one file, `telepathy.cpp`. It should implement the following function. The program should include `telepathy.h` using the preprocessing directive `#include`.

- `vector<int> Aitana(int N, vector<int> A, vector<int> B, int S, int subtask)`
  This function implements Aitana's strategy. This function is called exactly once for each scenario, totaling Q times.
  - The parameter `N` is the number of sites in the national park N.
  - The parameters `A` and `B` are arrays of length $N - 1$, and for each j ($0 \leq j \leq N - 2$), `A[j]` and `B[j]` are the labels of two sites that a road connects, A_j and B_j.
  - The parameter S is the label of Aitana's current site.
  - The parameter `subtask` is the subtask number of the testcase, which is any of $1, 2, 3$.
  - The return value is the array $[x_0, x_1, \ldots, x_{10N}]$, which represents Aitana's movement.
  - The length of the return value's array must be exactly $10N + 1$. If not, your program will be judged as **Wrong Answer [1]**.
  - For each k ($0 \leq k \leq 10N$), $0 \leq x_k \leq N - 1$ must hold. If not, your program will be judged as **Wrong Answer [2]**.
  - $x_0 = S$ must hold. If not, your program will be judged as **Wrong Answer [3]**.
  - For each k ($1 \leq k \leq 10N$), the sites labeled x_{k-1} and x_k must be the same or are directly connected by the road. If not, your program will be judged as **Wrong Answer [4]**.

  Note that the "label" in the explanation above is based on Aitana's labeling.
- `vector<int> Bruno(int N, vector<int> C, vector<int> D, int T, int subtask)`
  This function implements Bruno's strategy. This function is called exactly once for each scenario, right after the call of the function `Aitana`, totaling Q times.
  - The parameter `N` is the number of sites in the national park N.
  - The parameters `C` and `D` are arrays of length $N - 1$, and for each j ($0 \leq j \leq N - 2$), `C[j]` and `D[j]` are the labels of two sites that a road connects, C_j and D_j.

  - The parameter T is the label of Bruno's current site.
  - The parameter `subtask` is the subtask number of the testcase, which is any of $1, 2, 3$.
  - The return value is the array $[y_0, y_1, \ldots, y_{10N}]$, which represents Bruno's movement.
  - The length of the return value's array must be exactly $10N + 1$. If not, your program will be judged as **Wrong Answer [5]**.
  - For each k ($0 \leq k \leq 10N$), $0 \leq y_k \leq N - 1$ must hold. If not, your program will be judged as **Wrong Answer [6]**.
  - $y_0 = T$ must hold. If not, your program will be judged as **Wrong Answer [7]**.
  - For each k ($1 \leq k \leq 10N$), the sites labeled y_{k-1} and y_k must be the same or are directly connected by the road. If not, your program will be judged as **Wrong Answer [8]**.
- Note that the "label" in the explanation above is based on Bruno's labeling.

If Aitana and Bruno do not meet within $10N$ turns, that is, for each k ($0 \leq k \leq 10N$), the site labeled x_k in Aitana's map and the site labeled y_k in Bruno's map are different, your program will be judged as **Wrong Answer [9]**.

Important Notices

- Your program can implement other functions for internal use, or use global variables. When it is graded, it will be executed as two processes of Aitana and Bruno, so the process of Aitana and the process of Bruno cannot share global variables.
- Your program must not use the standard input and the standard output. Your program must not communicate with other files by any method. However, your program may output debugging information to the standard error.

Compilation and Test Run

You can download an archive file containing the sample grader to test your program from the contest webpage. The archive file also contains a sample source file of your program.

The sample grader is the file `grader.cpp`. In order to test your program, put `grader.cpp`, `telepathy.cpp`, `telepathy.h` in the same directory, and run the following command to compile your programs.

```
g++ -std=gnu++20 -O2 -o grader grader.cpp telepathy.cpp
```

Instead, you may run `compile.sh` contained in the archive file. In this case, run the following command to compile your programs.

```
./compile.sh
```

When the compilation succeeds, the executable file `grader` is generated.

Note that the actual grader is different from the sample grader. The sample grader will be executed as a single process, which will read input data from the standard input and write the results to the standard output.

Input for the Sample Grader

The sample grader reads the input data in the following format from the standard input. Here, the scenarios are numbered from 0 to $Q - 1$. Additionally, subtask represents the subtask number of the testcase.

subtask

Q

(The data of scenario 0)

(The data of scenario 1)

$\vdots$

(The data of scenario $Q - 1$)

The data of each scenario must be given in the following format.

N

$u_0\ v_0$

$u_1\ v_1$

$\vdots$

$u_{N-2}\ v_{N-2}$

$p_0\ p_1\ \cdots\ p_{N-1}$

$q_0\ q_1\ \cdots\ q_{N-1}$

$s\ t$

For the meaning of each variable, please refer to the **Problem Details** section of the problem statement. **Note that the information on Aitana's map and Bruno's map is not directly inputted.**

The way of shuffling the roads for the argument of the functions `Aitana` and `Bruno` is determined by the pseudorandom numbers whose results do not change for different executions. In order to change the seed of pseudorandom numbers, run the sample grader with the first integer argument as follows:

```
./grader 20250615
```

Output of the Sample Grader

The sample grader outputs the following information to the standard output (quotes for clarity) for each scenario, totaling Q lines.

- If correct, it outputs the turn number k^* when Aitana and Bruno meet again, and the minimum number of roads to pass to move from Aitana's current site and Bruno's current site, d, in this order, in the format as "`Case #0: Accepted 5 2`".
- If incorrect, the sample grader writes its type as "`Case #0: Wrong Answer [1]`".

Notices for Grading

The inputs for the actual grader may not be decided before the program's execution, and they may be determined by the return values of the functions `Aitana` and `Bruno` in the previous scenarios.

Constraints

In this problem, you need to solve the problem for at most 201 scenarios, that is, $1 \leq Q \leq 201$. Each scenario satisfies the following constraints.

- $2 \leq N \leq 200$.
- $(p_0, p_1, \ldots, p_{N-1})$ is the permutation of the integers between 0 and $N-1$ (inclusive).
- $(q_0, q_1, \ldots, q_{N-1})$ is the permutation of the integers between 0 and $N-1$ (inclusive).
- $0 \leq u_j \leq N-1$ $(0 \leq j \leq N-2)$.
- $0 \leq v_j \leq N-1$ $(0 \leq j \leq N-2)$.
- It is possible to move from any site to any site by passing through some roads.
- $0 \leq s \leq N-1$.
- $0 \leq t \leq N-1$.
- $s \neq t$.

Subtasks

1. (40 points) $(p_0, p_1, \ldots, p_{N-1}) = (q_0, q_1, \ldots, q_{N-1}) = (0, 1, \ldots, N-1)$.
2. (40 points) $u_j = j$, $v_j = j+1$ $(0 \leq j \leq N-2)$.

3. (20 points) No additional constraints.

Note that in subtask 3, the testcases where the parameter `subtask` (in the functions `Aitana` and `Bruno`) is 1 or 2 are also subject to grading.

Scoring Details

Let k^* be the turn number when Aitana and Bruno meets again, and let d be the minimum number of roads to pass to move from Aitana's current site and Bruno's current site. Additionally, let α be the maximum value of $\frac{k^*}{d}$ across all scenarios of the subtask.

If your program is judged as incorrect for some scenarios in the subtask, your score for this subtask is 0 points. Otherwise, you will get the following percentage of score for this subtask (if multiple conditions are met, the highest one is applied).

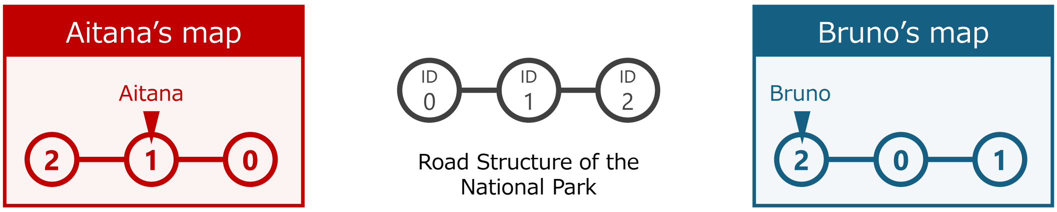
- If $k^* \leq 10N$ for all scenarios, 15 percent
- If $k^* \leq \max(10d, N)$ for all scenarios, 25 percent
- If $k^* \leq 10d$ for all scenarios:
  - If $9 < \alpha \leq 10$, 40 percent
  - If $6 < \alpha \leq 9$, $\lfloor 100 - 20(\alpha - 6) \rfloor$ percent
  - If $\alpha \leq 6$, 100 percent

Sample Communication

Here is a sample input for the sample grader, along with the corresponding function calls. Note that in the following table, some parts of the return values are omitted, but each represents an array of length 31.

Sample Input 1	Sample Function Calls	
	Calls and return values of `Aitana`'s	Calls and return values of `Bruno`
`2`	`Aitana(3, [0, 1], [1, 2], 1, 2)`	
`1`	`[1, 0, 0, 1, 2, ..., 2]`	
`3`		`Bruno(3, [1, 0], [2, 0], 2, 2)`
`0 1`		`[2, 2, 0, 0, 1, ..., 1]`
`1 2`		
`2 1 0`		
`2 0 1`		
`1 0`		

In this example, the road structure of the national park and Aitana's and Bruno's maps are represented in the following figure.



Aitana moves in a way that, for each turn, she is at sites labeled $1, 0, 0, 1, 2, \ldots, 2$ (in her map), respectively. Bruno moves in a way that, for each turn, he is at sites labeled $2, 2, 0, 0, 1, \ldots, 1$ (in his map), respectively. They meet again when the 3-rd turn is finished. At this moment, Aitana is at the site labeled 1 (on her map), and Bruno is at the site labeled 0 (on his map), which indicates the same place.