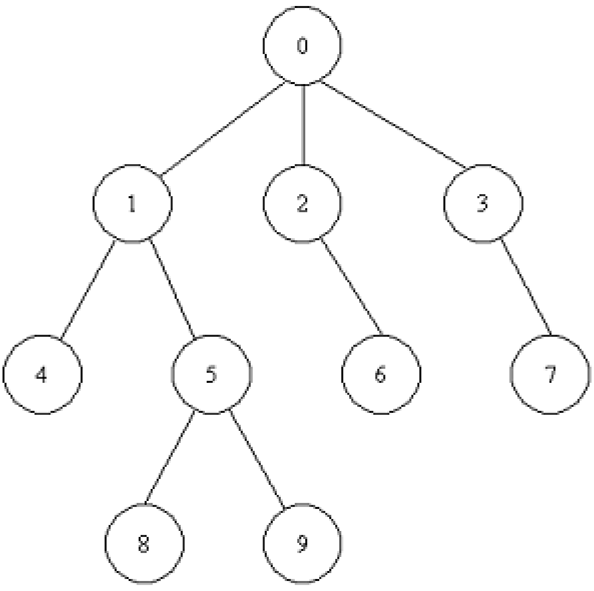
Scientists are conducting research on the behavior of a newly discovered Agamic Cellular Microbe. This special kind of microbe is capable of massively reproducing by itself in a short time. The lifetime of an ACM consists of three phases:

1. The infancy phase, which starts from its birth and lasts for approximately several seconds;
2. The multiplication phase, in which one ACM can procreate up to 100 offspring in only several milliseconds;
3. The mature phase, in which it remains inactive for the rest of its life.

At the beginning of the experiment, a newborn, single cell of ACM, is put into a suitable circumstance for its production. This cell, numbered as 0, starts to multiply and its descendants are numbered, starting from 1, according to their positions in the family hierarchy. During the experiment special equipment is used to record the numbers of the offspring generated by each of the ACM's. The experiment is stopped after a certain time period.



The family tree of ACM's in the first case of sample input

Your task is to help the scientists to determine whether one ACM is an ancestor of another.

Input

Standard input will contain multiple test cases. The first line of the input is a single integer T ($1 \leq T \leq 10$) which is the number of test cases. T test cases follow, each preceded by a single blank line.

Each test case starts with a single integer N ($1 \leq N \leq 300,000$) which is the number of ACM's that have their descendants recorded. The following N integers (not necessarily on a same line), C_i ($0 \leq i < N, 0 \leq C_i \leq 100$), give the number of offspring of the i-th ACM. The next line contains an integer M ($1 \leq M \leq 1,000,000$) which is the number of queries. M lines follow, each contains two integers a and b, querying whether the a-th ACM is an ancestor of the b-th ACM.

The total number of ACM's may be greater than N, but would never exceed 20,000,000.

Output

Results should be directed to standard output. Start each case with '`Case #:`' on a single line, where # is the case number starting from 1. Two consecutive cases should be separated by a single blank line. No blank line should be produced after the last test case.

For each query, print either '`Yes`' or '`No`' on a single line, which is the answer to the query.

Sample Input

```
2

6
3 2 1 1 0 2
5
0 1
2 4
3 5
1 8
6 9

5
2 0 3 0 1
4
2 6
1 6
2 3
3 5
```

Sample Output

```
Case 1:
Yes
No
No
Yes
No

Case 2:
Yes
No
Yes
No
```