G. Master Cat Viper and Programming

Score: 1

CPU: 2s
Memory: 1024MB

In the city of Zou, there are n houses. The houses are connected by bidirectional roads in such a way that they form a tree i.e., there is exactly one path between each pair of houses. The houses are numbered from 1 to n. The king of Zou, Master Cat Viper has recently learnt programming. Now he wants to calculate the closeness between two houses. He defines the closeness between two houses u and v as follows:

1. Let the house numbers that lie between the path from house u to house v are u, a_1, a_2, …, v and make an array A with these values where A[0] = u, A[1] = a_1, A[2] = a_2, …, A[m-1] = v. (assuming there are m houses that lie between the path from u to v)
2. Call squareSum(A[], m) to calculate the closeness. The squareSum function is defined below:

```
long long int squareSum(int A[], int m)
{
    long long int mod = 1000000007;
    long long int closeness = 0;
    for(int i = 0; i < m; i++)
    {
        for(int j = i+1; j < m; j++)
        {
            long long int tmp = A[i]+A[j];
            closeness = (closeness+tmp*tmp)%mod;
        }
    }
    return closeness;
}
```

The king knows that the proposed algorithm is not an efficient one. So he hires you to do the task. Given the description of the city, the king is going to ask you some queries where each query consists of two house numbers and you need to find the closeness between them.

Input

Input starts with an integer **T** denoting the number of test cases.
The first line of each case contains two integers **n** and **Q**, denoting the number of houses and the number of queries, respectively.
The next line contains **n** space separated integers **p_1, p_2, p_3, …, p_n** where **p_i** means there is a bidirectional road between houses **i** and **p_i**. **p_i = 0** means there is no road. It is guaranteed that among all **p_i** values, only one will be **0**.
Then follows **Q** lines, each containing a pair of integers **u** and **v** (**u != v**) which means you need to calculate the closeness between houses **u** and **v**.

Constraints

$1 \le T \le 10$
$2 \le n \le 50000$
$1 \le Q \le 50000$
$1 \le u, v \le n$
$0 \le p_i \le n$

Output

For each case, print the case number first. Then for each query, print the closeness value between the given houses. See sample I/O for more clarification.

Sample

Input	Output

`1`
`4 3`
`4 1 2 0`
`1 3`
`3 4`
`2 3`
	`Case 1:`
	`50`
	`160`
	`25`