Problem C
Conservation

The most famous painting in Byteland—a portrait of a lady with a computer mouse by Leonardo da Bitci—needs to be conserved. The work will be conducted in two narrowly specialized laboratories. The conservation process has been divided into several stages. For each of them, we know the laboratory in which it will take place.

Transporting the very precious and fragile painting introduces additional risk; therefore, it should be avoided whenever possible. Ideally, all the work in the first laboratory would be done, and then the painting would be moved to the second one. Unfortunately, there are several dependencies between the conservation stages—some of them need to be completed before others may begin. Your task is to find an ordering of conservation stages that minimizes the number of times the painting needs to be moved from one laboratory to the other. The conservation can begin in any of the two laboratories.

Input

The first line of the input contains the number of test cases T, where $1 \leq T \leq 35\,000$. The descriptions of the test cases follow:

The first line of each test case contains two space-separated integers n and m ($1 \leq n \leq 100\,000$, $0 \leq m \leq 1\,000\,000$)—the number of conservation stages and the number of dependencies between them. In the next line there are n space-separated integers—the i-th of them is 1 if the i-th conservation stage will take place in the first laboratory, and 2 otherwise. The following m lines contain pairs of integers i, j ($1 \leq i, j \leq n$), denoting that the i-th stage has to be completed before the j-th. The sum of m overall T test cases is at most $3 \cdot 10^6$.

You may assume that it is always possible to order the conservation stages so that all the dependencies are satisfied.

Output

Print the answers to the test cases in the order in which they appear in the input. For each test case, output a single line containing the minimal number of times the painting needs to be transported between the laboratories.

Sample Input 1

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

Sample Output 1

```
2
```