GFSSOC '15 Fall S5 - Raider

Time limit: 2.5s **Memory limit:** 256M

Calvin has recently programmed a game and wants you to beta test it for him. In this game, you are playing as an adventurer in a world of N towns (numbered from 1 to N) and M unidirectional roads that run between them. In each of these towns, there is a bank you can borrow stuff from, and V_i coins can be borrowed from each bank i. Your job is to try and borrow as many coins as possible! You may visit towns as many times as you want, but the bank does not refill after you take a loan from them.

In addition, we know there are a certain number of provinces in this world. A province is defined as a maximal group of cities in which you can travel from any city in the group to any other city in the group by following some roads. Calvin has made this game more fun by programming an anti-borrow system. You CANNOT borrow from two consecutive provinces on your path as men in black suits will show up and confiscate your loot. Being allergic to losing, you would like to find the maximum number of coins you can borrow in one run of this game from town 1 to town N, and the number of distinct paths you can take to do this. A distinct path is defined as a path **of provinces** that either goes to provinces in a different order, visits a different set of provinces, or borrows from a different set of provinces. More formally, two paths A and B consisting of a set of provinces are considered distinct only if $|A| \neq |B|$ or $A_i \neq B_i$ for any i $(1 \leq i \leq |A|)$. Since there may be many ways, output the number of ways modulo $10^9 + 7$.

Input Specification

Line 1: 2 integers, space separated — N and M $(2 \leq N \leq 500\,000, 1 \leq M \leq 1\,500\,000)$

Next line: N space separated integers, the i^{th} integer is the amount of loot you can hoard from town i, V_i $(0 \leq V_i \leq 1\,000)$

Next M lines: 2 integers, space separated — a and b. Each pair indicates a unidirectional path from town a to town b $(1 \leq a, b \leq N)$. There may be self-loops and duplicate edges.

Output Specification

One line with two integers space separated, the maximum amount you can borrow and the number of ways there are to do this. It is guaranteed that there is at least one way to borrow from town 1 to town N.

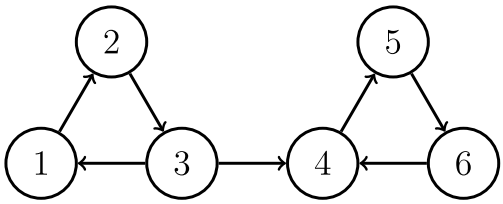
Sample Input

```
6 7
1 1 1 1 1 1
1 2
2 3
3 1
4 5
5 6
6 4
3 4
```

Sample Output

```
3 2
```

Explanation of Output for Sample Input



The towns are arranged as shown above, and each town can be looted for 1 unit of money. The sets of towns $\{1, 2, 3\}$ and $\{4, 5, 6\}$ are the provinces, since every town leads to every other town in each set. You want to get from town 1 to 6, and cannot visit both provinces since they are connected with a path, or else you will get beat up. Therefore, the best paths to take is travelling through all the towns, either looting all banks in province $\{1, 2, 3\}$, or looting all banks in the province $\{4, 5, 6\}$. After doing this, you can take the remaining paths to arrive at town 6.