20-21 своя тренировка 3: Graphs

A. Just Counting

1 second, 512 mebibytes

You are given an undirected graph without loops and multiple edges.

Find the number of ways to write integers $[0; 4]$ on edges such that for each vertex, the sum of weights of edges incident to it will be equal to zero modulo five (i.e. is equal to $5k$ for some integer k).

As the answer may be very large, you only need to find it modulo $998\,244\,353$.

Input

The first line of input contains one integer t ($1 \le t \le 500\,000$): the number of testcases.

The next lines contain t descriptions of test cases.

The first line of each test case contains two integers n, m ($1 \le n \le 200\,000, 0 \le m \le 300\,000$): the number of vertices.

The next m lines contain descriptions of edges, where the i-th of them contains two integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$), denoting an edge connecting vertices a_i and b_i in the graph.

It is guaranteed that there are no multiple edges.

It is also **guaranteed** that the total sum of $n + m$ in all test cases is at most $500\,000$.

Output

For each test case, print one integer: the number of ways to write integers $[0; 4]$ on edges such that for each vertex, the sum of weights of edges incident to it will be equal to zero modulo five (i.e. is equal to $5k$ for some integer k), modulo $998\,244\,353$.

input

```
3
1 0
3 3
1 2
2 3
3 1
4 4
1 2
2 3
3 4
4 1
```

output

```
1
1
5
```

B. Colorful Tree

2 seconds, 128 megabytes

There is a tree having n nodes, the i-th node of which has a type of color, denoted by an integer c_i.

The path between every two nodes is unique, of which we define the value is the number of distinct types of colors appearing on it.

Calculate the sum of values of all possible paths, $\frac{n(n-1)}{2}$ in total, between two different nodes on the tree.

Input

The input contains multiple (about 50) test cases.

For each test case, the first line contains an integer n ($2 \le n \le 2 \times 10^5$), indicating the number of nodes.

The next line contains n integers, the i-th number of which is c_i ($1 \le c_i \le n$), denoting the color of the i-th node.

Each of the next $(n-1)$ lines contains two integers x, y ($1 \le x, y \le n$, $x \ne y$), representing an edge between the x-th node and the y-th one. It is guaranteed that given edges form a tree.

Output

For each test case, output "`Case #x: y`" in one line (without quotes), where x indicates the case number starting from 1, and y denotes the answer to the corresponding case.

input

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

output

```
Case #1: 6
Case #2: 29
```

C. Taiga Tree

1 second, 256 megabytes

You have a tree, or an undirected connected graph with no cycles, with n vertices and $n-1$ edges. Vertex 1 is the root.

You define a "leaf vertex" to be a vertex on the tree, other than the root, that is adjacent to exactly one branch vertex.

You also define a "branch vertex" to be a vertex on the tree other than the root, that is adjacent to exactly two other vertices, and adjacent to at least one leaf vertex.

You define a tree to be more of a "taiga tree" the more branch vertices that it has. Given a tree, figure out how many branch vertices it has.

Input

The first line of input contains a single positive integer n $(1 <= n <= 10^5)$: the number of vertices on the tree.

The next $n-1$ lines each contain two space-separated integers, each representing an edge on the tree.

Output

Output a single positive integer: the number of branch vertices on the tree, as defined above.

Scoring

Full problem: 15 points

input

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

output

```
2
```

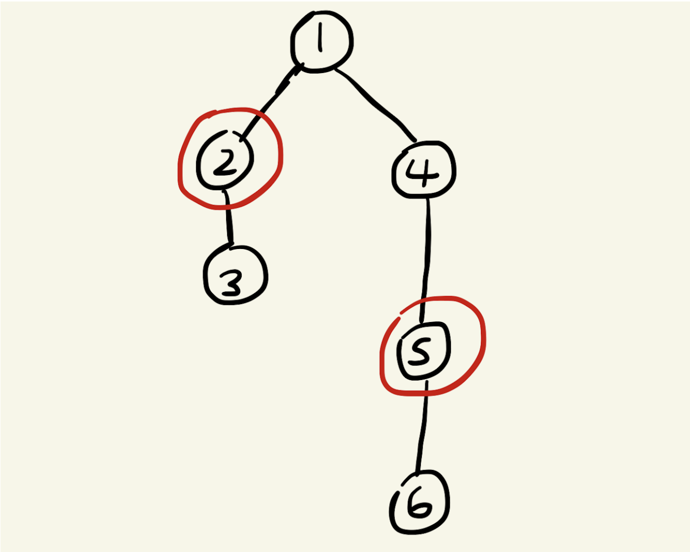
input

```
8
1 2
2 3
1 4
4 5
5 6
6 7
6 8
```

output

```
1
```

Here is a visual representation of the first example case (the branch vertices are circled):



D. Xor the graph

1 second, 256 megabytes

You are given an undirected graph with n nodes and m edges.

The graph doesn't contain self-loops but it may contain multiple edges.

There is a number a_i attached to the i_{th} $(1 \le i \le n)$ node.

You can do the following operation once: Choose a set of nodes and a value x $(0 \le x < 2^{20})$ and change all the values of the nodes in the set from a_i into $a_i \bigoplus x$.

You should choose any set and any value x so that for each edge the values of the nodes connected with that edge are different.

Is it possible?

Input

The first line of input contains two integers n and m, which are the number of nodes and the number of edges $(1 \le n, m \le 3 \times 10^5)$.

The second line contains n integers, the i^{th} one is a_i which is the value attached to the i^{th} node $(0 \le a_i < 2^{20})$.

The next m lines will contain two integers for each u and v, $(1 \le u, v \le n)$ $(u \ne v)$, which means that there is an edge between nodes u and v.

it is guaranteed that the given graph doesn't contain self-loops but it may contain multiple edges.

Output

If there is no way to choose a set and a value x, print -1.

Otherwise print two integers k and x on the first line, which is the size of the chosen set and the chosen value, $(1 \le k \le n)$ $(0 \le x < 2^{20})$.

In the second line print k integers, which describes the chosen nodes in the set.

Make sure that no node appears more than one time in the set.

input

```
3 3
1 1 1
1 2
2 3
1 3
```

output

```
-1
```

input

```
3 3
1 1 2
1 2
2 3
1 3
```

output

```
1 1
2
```

input

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

output

```
0 1
```

E. Guarding the Temples

2 seconds, 64 megabytes

There are thousands of Buddhist temples in Thailand, known as "wat". Some of them, such as the "Wat Phra Kaew" in Bangkok's Grand Palace, are specially regarded for their importance and they're called royal temples. The "Wat Phra Kaew" is famous for housing the Emerald Buddha statue, a national treasure. In 2016, the ACM ICPC World Finals will take place in Phuket, Thailand, and so increased tourism is expected in this city. The authorities in Phuket thus want to improve security in its royal temples.

Phuket's Security Unit (PSU) hired the researcher Lua "the ingenious" Kuratowski. PSU needs to solve the following problem. Given N royal temples and M streets connecting them, they must position guards at these streets so that every royal temple can be under surveillance. They consider a temple as secured if there is a guard in at least one of the streets ending at the temple. Streets were laid out so that one can reach any temple from any other. Moreover, due to a cultural dislike of odd numbers, any time someone starts a walk at a temple and returns to it, the number of streets traversed is always even.

Lua knows you wish to advance to the World Finals next year, and she considers this to be a good test for your skills. She challenges you to find a solution with the minimum number of guards.

Input

The first line has a single integer T, the number of test cases.

Each test case spans several input lines. The first line has two integers, N and M, the number of royal temples and the number of streets joining temples, respectively. Each temple is represented by an integer between 1 and N. The next M lines describe the streets. Each street is represented by two integers, corresponding to the temples it joins.

Limits

- $1 \le T \le 10$
- $1 \le N \le 10^3$
- $1 \le M \le 5 \cdot 10^3$

Output

For each instance, print a single line with the minimum number of guards needed to have all royal temples under surveillance.

input

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

output

```
3
3
```

F. Cutting a sheet

2 s., 256 MB

You've gotten an $n \times m$ sheet of checkered paper. Some of its cells are painted. The set of all colored cells of a sheet of paper will be denoted by A. The set A is connected. It is required to find the minimum number of cells that can be removed from the set A so that it ceases to be connected.

A set of painted cells is called *connected* if for every two cells a and b from this set there is a sequence of cells in the set, starting in a and ending in b, such that any cell of this sequence, excluding the last one, has a common side with the next cell in the sequence. An empty set and a set consisting of one cell are considered connected by definition.

Input

The first input line contains two space-separated integers n and m ($1 \le n, m \le 50$) — the sizes of the sheet of paper.

Each of the next n lines contains m characters — the description of the sheet of paper: the j-th character of the i-th line equals either "#", if the corresponding cell is painted (belongs to set A), or equals "." if the corresponding cell is not painted (does not belong to set A). It is guaranteed that the set of all painted cells A is connected and isn't empty.

Output

On the first line print the minimum number of cells that need to be deleted to make set A not connected. If it is impossible, print -1.

input

```
5 4
####
#..#
#..#
#..#
####
```

output

```
2
```

input

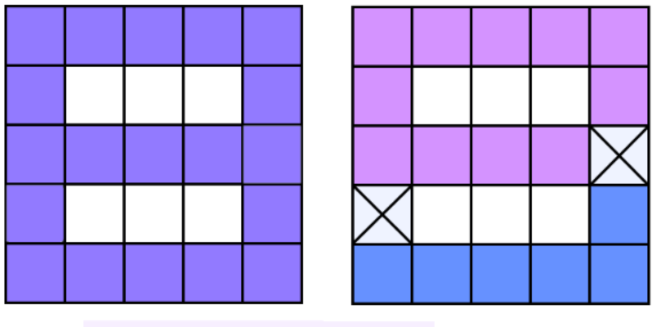
```
5 5
#####
#...#
#####
#...#
#####
```

output

```
2
```

In the first sample you can delete any two cells that do not share a side. After that the set of painted cells is not connected anymore.

The note to the second sample is shown on the figure below. To the left there is a picture of the initial set of cells. To the right there is a set with deleted cells. The deleted cells are marked with crosses.



G. Dissatisfaction of motorists

4 s., 256 MB

One country has n cities and m two-way roads. Each road connects a pair of cities, and for each road the level of dissatisfaction with it by motorists is known — the value w_i.

For each road we know the value c_i — how many tugriks we should spend to reduce the level of dissatisfaction with this road by one. Thus, to reduce the dissatisfaction with the i-th road by k, we should spend $k \cdot c_i$ tugriks. And **it is allowed for the dissatisfaction to become zero or even negative**.

In accordance with the president's decree, we need to choose $n - 1$ roads and make them the *main roads*. An important condition must hold: it should be possible to travel from any city to any other by the *main roads*.

The road ministry has a budget of S tugriks for the reform. The ministry is going to spend this budget for repair of some roads (to reduce the dissatisfaction with them), and then to choose the $n - 1$ *main roads*.

Help to spend the budget in such a way and then to choose the main roads so that the total dissatisfaction with the *main roads* will be as small as possible. The dissatisfaction with some roads can become negative. It is not necessary to spend whole budget S.

It is guaranteed that it is possible to travel from any city to any other using existing roads. Each road in the country is a two-way road.

Input

The first line contains two integers n and m ($2 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le 2 \cdot 10^5$) — the number of cities and the number of roads in the country, respectively.

The second line contains m integers $w_1, w_2, \dots, w_m$ ($1 \le w_i \le 10^9$), where w_i is the drivers dissatisfaction with the i-th road.

The third line contains m integers $c_1, c_2, \dots, c_m$ ($1 \le c_i \le 10^9$), where c_i is the cost (in tugriks) of reducing the dissatisfaction with the i-th road by one.

The next m lines contain the description of the roads. The i-th of this lines contain a pair of integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) which mean that the i-th road connects cities a_i and b_i. All roads are two-way oriented so it is possible to move by the i-th road from a_i to b_i, and vice versa. It is allowed that a pair of cities is connected by more than one road.

The last line contains one integer S ($0 \le S \le 10^9$) — the number of tugriks which we can spend for reforms.

Output

In the first line print K — the minimum possible total dissatisfaction with *main roads*.

In each of the next $n - 1$ lines print two integers x, v_x, which mean that the road x is among main roads and the road x, after the reform, has the level of dissatisfaction v_x.

Consider that roads are numbered from 1 to m in the order as they are given in the input data. The edges can be printed in arbitrary order. If there are several answers, print any of them.

input

```
6 9
1 3 1 1 3 1 2 2 2
4 1 4 2 2 5 3 1 6
1 2
1 3
2 3
2 4
2 5
3 5
3 6
4 5
5 6
7
```

output

```
0
1 1
3 1
6 1
7 2
8 -5
```

input

```
3 3
9 5 1
7 7 2
2 1
3 1
3 2
2
```

output

```
5
3 0
2 5
```

H. Array and operations

1 s., 256 MB

You wrote down on a piece of paper an array of n positive integers $a[1], a[2], \dots, a[n]$ and m *good* pairs of integers $(i_1, j_1), (i_2, j_2), \dots, (i_m, j_m)$. Each *good* pair (i_k, j_k) satisfies the conditions: $i_k + j_k$ is an odd number and $1 \le i_k < j_k \le n$.

In one operation you can perform a sequence of actions:

- take one of the *good* pairs (i_k, j_k) and some integer v ($v > 1$), which divides both numbers $a[i_k]$ and $a[j_k]$;
- divide both numbers by v, i. e. perform the assignments: $a[i_k] = \frac{a[i_k]}{v}$ and $a[j_k] = \frac{a[j_k]}{v}$.

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

Input

The first line contains two space-separated integers n, m ($2 \le n \le 100$, $1 \le m \le 100$).

The second line contains n space-separated integers $a[1], a[2], \dots, a[n]$ ($1 \le a[i] \le 10^9$) — the description of the array.

The following m lines contain the description of *good* pairs. The k-th line contains two space-separated integers i_k, j_k ($1 \le i_k < j_k \le n$, $i_k + j_k$ is an odd number).

It is guaranteed that all the *good* pairs are distinct.

Output

Output the answer for the problem.

input

```
3 2
8 3 8
1 2
2 3
```

output

```
0
```

input

```
3 2
8 12 8
1 2
2 3
```

output

```
2
```

I. Door problem

2 s., 256 MB

Loki has trapped n people in n distinct rooms in a hotel.The doors to some rooms are locked, others are unlocked. However, there is a condition that people can only escape when all the doors are unlocked at the same time. There are m switches. Each switch control doors to some rooms, however each door is controlled by **exactly two** switches.

You are given the initial configuration of the doors. Toggling any switch, that is, turning it ON when it is OFF, or turning it OFF when it is ON, toggles the condition of the doors that this switch controls. Say, we toggled switch 1, which was connected to room 1, 2 and 3 which were respectively locked, unlocked and unlocked. Then, after toggling the switch, they become unlocked, locked and locked.

You need to tell Sherlock, if there exists a way to unlock all doors at the same time.

Input

First line of input contains two integers n and m ($2 \le n \le 10^5$, $2 \le m \le 10^5$) — the number of rooms and the number of switches.

Next line contains n space-separated integers $r_1, r_2, \dots, r_n$ ($0 \le r_i \le 1$) which tell the status of room doors. The i-th room is locked if $r_i = 0$, otherwise it is unlocked.

The i-th of next m lines contains an integer x_i ($0 \le x_i \le n$) followed by x_i distinct integers separated by space, denoting the number of rooms controlled by the i-th switch followed by the room numbers that this switch controls. It is guaranteed that the room numbers are in the range from 1 to n. It is guaranteed that each door is controlled by exactly two switches.

Output

Output "YES" without quotes, if it is possible to open all doors at the same time, otherwise output "NO" without quotes.

input

```
3 3
1 0 1
2 1 3
2 1 2
2 2 3
```

output

```
NO
```

input

```
3 3
1 0 1
3 1 2 3
1 2
2 1 3
```

output

```
YES
```

input

```
3 3
1 0 1
3 1 2 3
2 1 2
1 3
```

output

```
NO
```

In the second example input, the initial statuses of the doors are $[1, 0, 1]$ (0 means locked, 1 — unlocked).

After toggling switch 3, we get $[0, 0, 0]$ that means all doors are locked.

Then, after toggling switch 1, we get $[1, 1, 1]$ that means all doors are unlocked.

It can be seen that for the first and for the third example inputs it is not possible to make all doors unlocked.

J. Seras, Orican and the bipartiteness

2 s., 256 MB

Seras and Orican continue their adventures. Every inhabitant of the Necron empire knows that the Silent King loves bipartite graphs, especially trees.

A tree is a connected acyclic graph. A bipartite graph is a graph, whose vertices can be partitioned into 2 sets in such a way, that for each edge (u, v) that belongs to the graph, u and v belong to different sets. You can find more formal definitions of a tree and a bipartite graph in the notes section below.

Silent King gave Seras and Orican a tree consisting of n nodes and asked them to add edges to it in such a way, that the graph is still bipartite. Besides, after adding these edges the graph should be simple (doesn't contain loops or multiple edges). What is the maximum number of edges they can add?

A loop is an edge, which connects a node with itself. Graph doesn't contain multiple edges when for each pair of nodes there is no more than one edge between them. **A cycle and a loop aren't the same** .

Input

The first line of input contains an integer n — the number of nodes in the tree ($1 \le n \le 10^5$).

The next $n - 1$ lines contain integers u and v ($1 \le u, v \le n$, $u \ne v$) — the description of the edges of the tree.

It's guaranteed that the given graph is a tree.

Output

Output one integer — the maximum number of edges that Seras and Orican can add to the tree while fulfilling the conditions.

input

```
3
1 2
1 3
```

output

```
0
```

input

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

output

```
2
```

Tree definition: https://en.wikipedia.org/wiki/Tree_(graph_theory)

Bipartite graph definition: https://en.wikipedia.org/wiki/Bipartite_graph

In the first test case the only edge that can be added in such a way, that graph won't contain loops or multiple edges is $(2, 3)$, but adding this edge will make the graph non-bipartite so the answer is 0.

In the second test case Seras and Orican can add edges $(1, 4)$ and $(2, 5)$.

K. Fulgrim's peach-tree

1 s., 256 MB

In Fulgrim's garden there grows a peculiar peach-tree that fruits one time per year. Its peculiarity can be explained in following way: there are n inflorescences, numbered from 1 to n. Inflorescence number 1 is situated near base of tree and any other inflorescence with number i ($i > 1$) is situated at the top of branch, which bottom is p_i-th inflorescence and $p_i < i$.

Once tree starts fruiting, there appears exactly one peach in each inflorescence. The same moment as peach appear, they start to roll down along branches to the very base of tree. Each second all peaches, except ones in first inflorescence simultaneously roll down one branch closer to tree base, e.g. peach in a-th inflorescence gets to p_a-th inflorescence. Peaches that end up in first inflorescence are gathered by Fulgrim in exactly the same moment. Second peculiarity of this tree is that once two peaches are in same inflorescence they **annihilate**. This happens with each pair of peaches, e.g. if there are 5 peaches in same inflorescence in same time, only one will not be annihilated and if there are 8 peaches, all peaches will be annihilated. Thus, there can be no more than one peach in each inflorescence in each moment of time.

Help Fulgrim with counting number of peaches he will be able to collect from first inflorescence during one harvest.

Input

First line of input contains single integer number n ($2 \le n \le 100\,000$) — number of inflorescences.

Second line of input contains sequence of $n - 1$ integer numbers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$), where p_i is number of inflorescence into which the peach from i-th inflorescence rolls down.

Output

Single line of output should contain one integer number: amount of peaches that Fulgrim will be able to collect from first inflorescence during one harvest.

input

```
3
1 1
```

output

```
1
```

input

```
5
1 2 2 2
```

output

```
3
```

input

```
18
1 1 1 4 4 3 2 2 2 10 8 9 9 9 10 10 4
```

output

```
4
```

In first example Fulgrim will be able to collect only one peach, initially situated in 1st inflorescence. In next second peaches from 2nd and 3rd inflorescences will roll down and annihilate, and Fulgrim won't be able to collect them.

In the second example Fulgrim will be able to collect 3 peaches. First one is one initially situated in first inflorescence. In a second peach from 2nd inflorescence will roll down to 1st (Fulgrim will collect it) and peaches from 3rd, 4th, 5th inflorescences will roll down to 2nd. Two of them will annihilate and one not annihilated will roll down from 2-nd inflorescence to 1st one in the next second and Fulgrim will collect it.