**Embrace the Dark Side:**

Legend has it, there is field that is exactly like a matrix of size NxM. In a given location in this matrix-like field either a sith lord, a jedi warrior or a tree resides. Since the dark side is far more powerful than the light side, a sith lord can turn all adjacent jedi warrior into a sith lord in just 1 hour simultaneously. However, trees are very powerful beings and can never be turned into a sith lord.

After every hour, all jedi warriors that are adjacent to a sith lord turn into a sith lord themselves, and they too start converting other jedi warriors. What is the minimum time required to convert all jedi warriors into sith lords? If it is not possible to convert all jedi warriors print -1.

**Input format:**

First line contains two space separated integers N and M (rows and colums of the matrix)

Next N lines contain M space separated integer denoting the identity of the resident in the corresponding location in the matrix-like field (ith row, jth column represents the Matrix[i][j]) (0: tree, 1: jedi warrior, 2: sith lord)

**Output format:**

Print the minimum time required (in hours) to convert all jedi warriors into sith lords, if not possible print -1

**Constraints:**

1 <= N, M <= 10^3

**Manners:**

In the land of Sparta, the warrior class is highly respected among the society and there is a very strict hierarchy inside the warrior class. If a person is your commander, you have to wait him to eat before you can start eating. But also as an incentive, you have to give candy to the warriors below your command after you finish eating. Everyone has different amount generosity and a warrior gives candy equal to the number of candy that he has; however, if he has no candy he gives candy equal to his generosity to every inferior that he has.

There are N warriors tagged from 1 to N and it is dinner time. After all of the warriors had finished eating, the warriors who have no one in their command got together and counted their candies. How many candies does the warrior who has the most candies among the ones who got together have?

* No warrior is a commander of himself.
* There is no loop in commanding relation; more clearly, if ‘a’ commands ‘b’, ‘b’ commands ‘c’ then ‘c’ does not command ‘a’ (this is true for any length of chain).
* A warrior can have more than one commander.
* Everyone has zero candy in the beginning

**Input format:**

First line contains two space separated integers N and M: the number or warriors and the number of hierarchic relations.

Next line contains N space separated integers g1, …, gn denoting the generosity of the warriors

Next M lines contain two integers a\_i and a\_k: a\_i is a commander of a\_k

**Output format:**

Print the number of candies that the warrior with the most candies has (among the ones that has no one in their command)

**Constraints:**

1 <= N <= 2\*10^5

1 <= M <= 5\*10^5

1 <= a\_i, a\_k <= N

1 <= gi <= 100

**The Great Conqueror of Some Lands**

The great conqueror and mathematician Lambda has decided to inspect the land of Rome to see if it is a good land to conquer. Lambda has the map of Rome, this map consists of squares in a grid structure of size NxM and every square represents a land portion. Rome is a very organized land so every square is either a part of a city or a part of a forest. Adjacent city parts form a single city and adjacent forest parts form a single forest (diagonal adjacency does not count).

According to Lambda, a good land to conquer is defined like this:

* The land can be separated into states that has the same amount of cities in them (any subset of cities can constitute as a state).
* No state has less than two cities in it and there has to be more than one state.
* All cities have to be in exactly one state.

Even though Lambda is a very good mathematician he is busy so you have to determine if Rome is a good land to conquer or not. If it is good print “YES”, if not print “NO”, and then in the next line print the number of cities.

**Input format:**

First line contains two space separated integers N and M, size of the map of Rome

Next N lines contain M space separated integers r1, …, rm representing the grid map (0: forest part, 1: city part)

**Output format:**

Print “YES” or “NO” in the first line, then print the number of cities.

**Constraints:**

1 <= N, M <= 10^3

**Just Another Question**

You are given a rooted tree that has N number of nodes numbered from 1 to N and the root of the tree is the node 1. There are Q queries and for each query you are given an integer ‘c\_i’. Print the number of nodes in the subtree of the node ‘c\_i’ (the node ‘c\_i’ is included) for each query.

**Input format:**

First line contains two space separated integers N and Q, number of nodes in the tree, and number of queries.

Next N-1 lines contain two space separated integers ‘u\_i’ and ‘v\_i’, this means that there is an edge between node ‘u\_i’ and ‘v\_i’.

Next Q lines contain an integer ‘c\_i’, index of the node for each query.

**Output format:**

Print the number of nodes in the subtree of the given node in separate lines for each query

**Constraints:**

1 <= N, Q <= 10^6

1 <= c\_i <= N

**Requiem for a Pizza**

Master Yoda lives in a city where there is exactly one pizza place in every district. The city has N districts numbered from 1 to N, and there are two-way roads between districts (not necessarily between all pairs of districts but it is guaranteed that it is possible to go to every district from every district). There are M roads and all of the roads are 1 kilometer long.

Master Yoda wants to eat some pizza but he likes to work for it, so he does not want to buy pizza from a pizza place where the shortest distance from his house to that pizza place is less than L kilometers. He also does not want walk all day long, so he does not want to buy from a pizza place where the shortest distance from his house to that pizza place is more than R kilometers.

The price of pizza in different pizza places are different. What is the minimum amount of money that Master Yoda has to spend to buy pizza? If he cannot buy from any pizza place print -1.

* When counting the distances from his house to pizza places, Master Yoda only takes the roads into consideration, not inside district distances.
* Pizza places which are distance L and R from the house are fine to buy.

**Input format:**

First line contains three space separated integers N, M and H, number of districts, number of roads and district number of Master Yoda’s house.

Second line contains two space separated integers L and R, min and max distance that Master Yoda wants;

Third line contains N space separated integers p1, …, pn, prices of pizzas in pizza places.

Next M lines contain two space separated integers ‘u\_i’ and ‘v\_i’, this means that there is a two-way road between district ‘u\_i’ and ‘v\_i’.

**Output format:**

Print the minimum amount of money that Master Yoda has to spend to buy a pizza, print -1 if it is not possible to buy a pizza.

**Constraints:**

1 <= N, M <= 10^6

1 <= L <= R <=

1 <= ‘u\_i’, ‘v\_i’ <= N

1 <= pi <= 10^6