2022 Canadian Computing Olympiad
Day 2, Problem 1
Bi-ing Lottery Treekets

Time Limit: 1 second

Problem Description

In a parallel universe, everyone scored perfect on the CCO. As a result, Troy needs to pick the winner based on a lottery. Each contestant will choose numbers to create a ticket. A ticket is an array of size N indexed from 1 to N where each entry is a number from 0 to K.

The winning ticket is determined by dropping K balls (numbered from 1 to K) in a random sequence into a rooted binary tree. The tree has N nodes (numbered from 1 to N) and is rooted at node 1.

Each ball has a designated drop node that it will drop at. When a ball is dropped at an unoccupied node or enters an unoccupied node, one of three things happens:

1. If all of the current node's children are occupied by balls (or if a node has no children), the current ball rests at the current node. That is, it remains there and does not move again.
2. If the current node only has one unoccupied child, the current ball will move to this child.
3. If the current node has two unoccupied children, and if the current ball was just dropped, it could go either left or right. Otherwise, it will continue in the direction of its previous movement.

If all K balls cannot be dropped, a winning ticket is not determined. This happens when a ball is dropped and its drop node is occupied by another ball.

If all K balls have been dropped, the balls' resting positions determine the winning lottery ticket. The ith entry of the winning lottery ticket is the number of the ball that rests at node i or 0 if no ball rests at node i.

Troy would like to know the number of possible winning tickets (which could be zero).

Input Specification

The first line contains two space-separated integers N and K, denoting the number of nodes in the binary tree and the number of balls, respectively.

The next line contains K space-separated integers, where the ith integer denotes the designated drop node of the ball numbered i.

The last N lines each contain two space-separated integers. The ith line contains L_i and R_i denoting the ith node's left and right child, respectively, where 0 means no such child exists.

Marks Awarded	**Bounds on N**	**Bounds on K**	**Additional constraints**
3 marks	$1 \leq N \leq 12$	$1 \leq K \leq 6$	None
4 marks	$1 \leq N \leq 4000$	$1 \leq K \leq 4000$	All nodes do not have a left child
9 marks	$1 \leq N \leq 4000$	$1 \leq K \leq 4000$	The N drop nodes are distinct
9 marks	$1 \leq N \leq 4000$	$1 \leq K \leq 4000$	None

Output Specification

Output the remainder of the number of winning lottery tickets divided by $10^9 + 7$.

Sample Input 1

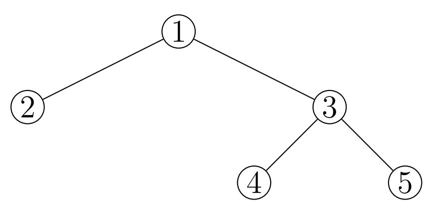
```
5 2
1 3
2 3
0 0
4 5
0 0
0 0
```

Output for Sample Input 1

```
4
```

Explanation of Output for Sample Input 1

The tree looks like this:



Consider when ball 1 is dropped first. If ball 1 goes left, then it will rest at node 2. Afterward, ball 2 is dropped and can rest at node 4 or 5. If ball 1 goes right, it will rest at node 5. Then, ball 2 will rest at node 4.

Consider when ball 2 is dropped first. Ball 2 can go left or right, resting at nodes 4 or 5, respectively. Then if ball 1 moves left after being dropped, it will rest at node 2. However, if ball 1 moves right, it will rest at either node 4 or 5, whichever place ball 2 does not occupy.

The possible winning tickets are [0, 1, 0, 2, 0], [0, 1, 0, 0, 2], [0, 0, 0, 1, 2], and [0, 0, 0, 2, 1].

Sample Input 2

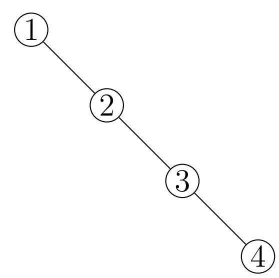
```
4 3
1 2 4
0 2
0 3
0 4
0 0
```

Output for Sample Input 2

```
2
```

Explanation of Output for Sample Input 2

The tree looks like this:



This test case satisfies the second subtask.

Ball 3 must be dropped first because if either ball 1 or ball 2 are dropped before ball 3, it would rest at node 4, which wouldn't allow ball 3 to be dropped.

Thus, we can either first drop ball 3, then ball 2 and finally ball 1 or we can first drop ball 3, then ball 1 and finally ball 2.

The possible winning tickets are [0,1,2,3] and [0,2,1,3].

Formal Definitions

A *binary tree* is a set of nodes that is either empty, or a root node with a left subtree and a right subtree both of which are binary trees. Given a node x, if its left subtree is not empty, then the root of that subtree is called the *left child* of x. Similarly, given a node x, if its right subtree is not empty, then the root of that subtree is called the *right child* of x.

2022 Canadian Computing Olympiad
Day 2, Problem 2
Phone Plans

Time Limit: 3 seconds

Problem Description

The mayor of CCOland, Jason, wants to install telephone lines amongst N households, which are numbered from 1 to N. To do so, he has asked two rivalling companies, Keenan Mobile Phones and Chris Home Telephone, for their phone plans. A phone plan for a company corresponds to a certain level and every telephone line has a level and company associated with it. If you have purchased a phone plan from a company with level l, then you are able to use all the telephone lines whose level is less than or equal to l that is associated with that company. A phone plan of level l costs $\$l$ and you cannot pick a phone plan of less than $\$0$.

Two households can only communicate with each other if they are connected by a path of telephone lines of the **same company**. Jason would like to buy one phone plan from each company of minimal cost such that there are at least K different pairs of households that can communicate with each other.

Input Specification

The first line contains four space-separated integers N, A, B and K, which represent the number of households, number of telephone lines from Keenan Mobile Phones, number of telephone lines from Chris Home Telephone and the minimum pairs of homes that need to be able to communicate with each other, respectively.

The next A lines each contain three space-separated integers u, v and l, which represents a Keenan Mobile Phones telephone line between household u and v ($1 \leq u, v \leq N$) that has a level l ($1 \leq l \leq 10^9$).

The next B lines have the same format as the previous A lines but for Chris Home Telephone.

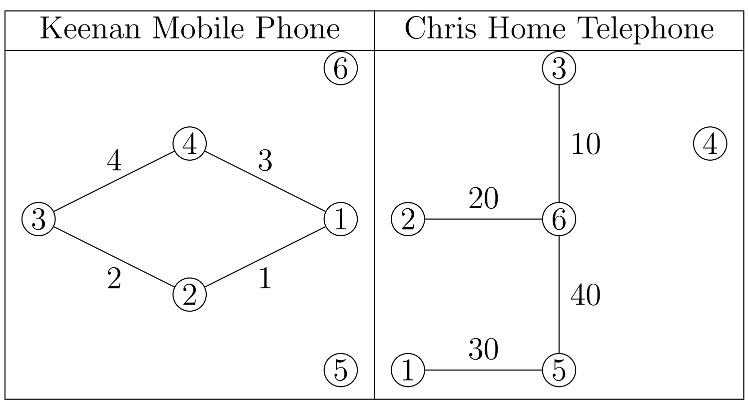
Marks Awarded	**Bounds on N**	**Bounds on A and B**	**Bounds on K**	**Additional Constraints**
6 marks	$1 \leq N \leq 2000$	$0 \leq A, B \leq 200\,000$	$0 \leq K \leq \frac{N(N-1)}{2}$	None
5 marks	$1 \leq N \leq 200\,000$	$0 \leq A, B \leq 200\,000$	$0 \leq K \leq \frac{N(N-1)}{2}$	Keenan Mobile Phones only connects to odd indexed households; Chris Home Telephone only connects to even indexed households
6 marks	$1 \leq N \leq 200\,000$	$0 \leq A \leq 10$; $0 \leq B \leq 200\,000$	$0 \leq K \leq \frac{N(N-1)}{2}$	None
8 marks	$1 \leq N \leq 200\,000$	$0 \leq A, B \leq 200\,000$	$0 \leq K \leq \frac{N(N-1)}{2}$	None

Output Specification
Output the cheapest cost needed to connect at least K different pairs of households or -1 if it is not possible.

Sample Input
```
6 4 4 9
1 2 1
2 3 2
1 4 3
3 4 4
5 6 40
1 5 30
2 6 20
3 6 10
```

Output for Sample Input
```
33
```

Explanation of Output for Sample Input
For each company, consider these pictures of the way the 6 households are connected by telephone lines:



If Jason buys phone plan level 3 from Keenan Mobile Phones and phone plan level 30 from Chris Home Telephone, then $(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)$ can communicate through Keenan Mobile Phones's lines and $(1,5),(2,6),(3,6),(2,3)$ can communicate through Chris Home Telephone's lines. There are no cheaper ways.

2022 Canadian Computing Olympiad
Day 2, Problem 3
Good Game

Time Limit: 1 second

Problem Description

Finn is playing a game of Twos and Threes. Twos and Threes is a one-player game played on a one-dimensional board. In the starting position, there are N blocks arranged in a row, with each block labelled either A or B. Blocks are numbered from 1 to N from left to right. Finn is allowed to make moves of the following form:

- Select 2 or 3 consecutive blocks that share the same label. Remove them from the board. Connect any remaining blocks together. Re-index the blocks from left to right starting with index 1.

Finn wins the game if all blocks are removed from the board. Your task is to help Finn determine a winning sequence of moves, or determine if the game cannot be won.

Input Specification

The first line of input will contain the integer N.

The second line of input will contain the string S which is the starting position of the game. There are N characters in S, and each of these characters in S is either A or B.

Marks Awarded	Bounds on N
3 marks	$1 \leq N \leq 15$
6 marks	$1 \leq N \leq 300$
7 marks	$1 \leq N \leq 6000$
9 marks	$1 \leq N \leq 10^6$

Output Specification

If there is a winning sequence of moves, output K, the number of moves in the winning sequence. On each of the next K lines, print an index i, followed by one space, followed by a number j, denoting a move that will remove the blocks currently at indices i to $i + j - 1$, inclusive.

If there is no winning sequence of moves, output -1.

If there are multiple winning sequences, then any winning sequence will be accepted. There is no need to minimize or maximize K.

Sample Input

```
9
ABAABBBAA
```

Possible Output for Sample Input

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

Explanation of Output for Sample Input

The sample output denotes this winning sequence:

$$ABAAB\underline{BB}AA$$
$$AB\underline{AA}BAA$$
$$A\underline{BB}AA$$
$$\underline{AAA}$$