

Sunday Drive

Input File: *drivein.txt*

Output File: *driveout.txt*

Time and Memory Limits: 1 second, 32 MB

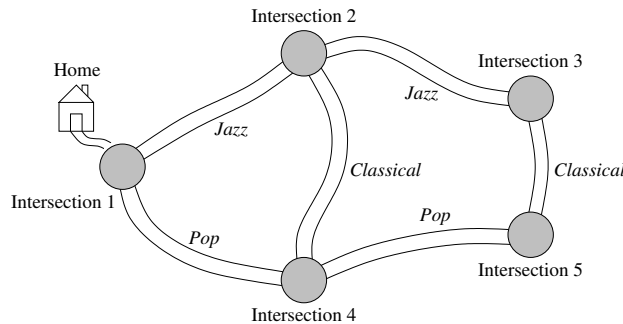
It is a bright Sunday morning, and your favourite thing to do on a Sunday morning is to drive around the town playing loud music with your car windows down. Unfortunately, not everybody in the town shares your taste in music.

The CD in your car contains S songs. Each song is of a particular style of music, such as jazz, pop or classical. Each street in the town has its own favourite style of music, and if you drive along a street playing the wrong style of music then the residents will throw eggs at your car. Clearly you would prefer this not to happen.

The town consists of N intersections and T streets, where each street joins two intersections. Each song on your CD plays for only one street—when you reach an intersection, the turning movement of the car makes the CD jump directly to the next song.

You wish to plan your drive so that you have as few eggs thrown at you as possible. Your drive must begin at your house, which lies at one of the intersections. Your drive must also end at your house, and it must pass through precisely S streets along the way. Streets may be used more than once; you may even make a U-turn at an intersection and drive back down the street along which you came (although you may not make a U-turn in the middle of a street).

As an example, consider the town illustrated below. There are $N = 5$ intersections in the town, and $T = 6$ streets joining them. Your house is at intersection 1.



Suppose your CD contains $S = 7$ songs, which have the following styles in order from first to last:

Jazz ; Classical ; Classical ; Pop ; Classical ; Pop ; Jazz

A good drive might be the following:

Song number	Street	CD style	Street style
1.	1–2	Jazz	Jazz
2.	2–4	Classical	Classical
3.	4–2	Classical	Classical
4.	2–3	Pop	Jazz
5.	3–5	Classical	Classical
6.	5–4	Pop	Pop
7.	4–1	Jazz	Pop

For songs number 1, 2, 3, 5 and 6, your song matches the favourite style of the street and the residents throw you flowers and kisses. For songs number 4 and 7, your song does not match and the residents throw eggs at your car.

Your task is to choose the S streets of your drive so that eggs are thrown at you as few times as possible.

Input

The first line of input will contain the single integer S , giving the number of songs on your CD ($1 \leq S \leq 500$).

The second line of input will describe the style of each song on the CD. For simplicity, each style is described by an integer in the range $1, 2, \dots, 20\,000$. The second line of input will therefore consist of S positive integers separated by spaces, giving the style of each song on the CD in order from the first song to the last song.

The third line of input will contain the two integers N and H separated by a single space, where N is the number of intersections ($1 \leq N \leq 200$) and H is the intersection at which you must start and finish ($1 \leq H \leq N$).

The fourth line of input will contain the single integer T , giving the number of streets ($0 \leq T \leq 20\,000$). Following this will be T additional lines, each describing a single street. Each of these lines will contain three integers $X Y M$ separated by single spaces, where the street joins intersections X and Y ($1 \leq X, Y \leq N$), and where M is the favourite style of music for the street ($1 \leq M \leq 20\,000$). No street will join an intersection with itself (that is, $X \neq Y$), and no pair of intersections will be joined by more than one street. All streets are two-way (i.e., you may drive along them in either direction).

It is guaranteed that it is possible to drive from intersection H back to intersection H again by following precisely S streets.

Output

Output must consist of a single integer, giving the smallest possible number of streets on which the residents throw eggs at you.

Sample Data

The following sample input and output describe the scenario illustrated earlier. Music styles 1, 2 and 3 correspond to jazz, classical and pop respectively.

Sample Input

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

Sample Output

```
2
```

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.