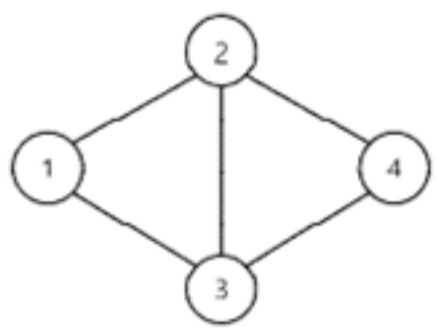
[HUST] Algorithm Problem Solving
Problem E : A distracted cat

Time Limit: 1000 ms Memory Limit: 64 MB

Problem

Chung keeps a cat. He created a three-dimensional playground for this cat, which has N rooms and M corridors. Rooms are numbered from 1 to N and one corridor connects two different rooms and can be moved in both directions. There is a maximum of one corridor that connects a pair of rooms. Any two rooms can be moved from one to the other using one or more corridors. The playground is three-dimensional, so no corridors cross. Below is an example with 4 rooms and 5 corridors.



The cats are very distracted and jump around the playground. Especially, k (≥ 3) different rooms (a1, a2, ..., ak) tend to be repeated and then ordered in the order. That is, it refers to the order of a1, a2, ..., ak, a1, a2, ..., ak, a1, a2, ... Of course, to do so, these rooms will have to be connected in a corridor in order. that is, a1 to a2 must be connected, a2 to a3 must be connected, ...,ak to a1.

Chung wants to prevent the cat from turning repeatedly because he is afraid of being too hard. To minimize the effort, I want to remove only one room (And the corridors connected to the room are closed.) so that there is no way the cat can spin repeatedly.

If the room is connected as in the previous example, it can be repeated in the order of rooms 1, 2, 3. Also, it can be repeatedly moved in the order of rooms 1, 2, 4, and 3. There is no way to turn repeatedly if you remove room number 2. If you remove room number 3, you get the same result. However, in the case of No. 4, there is still a way to turn it repeatedly, even if it is removed.

Takes the connection status of the rooms as input, and if a single room can be removed and the cats can eliminate all the repetitive ways, print out the room number. If there are several such rooms, print the sum of the numbers of those rooms. If there is no such room, 0 is output.

Input format

The following information is given as standard input. In the first line, an integer N (2≤N≤300,000) representing the number of rooms and an integer M (1≤M≤300,000) representing the number of corridors are given. Each of the following M lines is given a number of two different rooms connected by a corridor. It is ensured that there is at least one way in the rooms and corridors given as input, which is iterative.

Output format

With standard output, remove a single room and print out the number of the room if the cat can eliminate all the repetitive ways. If there are several such rooms, print the sum of the numbers of those rooms. If there is no such room, print 0.

Constraints of sub-problems

- Sub problem 1: 12 out of 100 points of the total score, N ≤ 5,000, M ≤ 5,000.
- Sub problem 2: 10 out of 100 points and M = N.
- Sub problem 3: 23 out of 100 points of the total score, and there are both the first room and the second room, the second room and the third room, the third room and the fourth room, ..., the hall connecting the room N and the first room.
- Sub problem 4: 55 out of 100 total points and there are no constraints other than the original constraints.

Input example

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

Output example

```
5
```

Input example

```
8 10
5 4
4 3
3 5
1 2
1 5
2 3
5 6
6 7
7 4
8 6
```

Output example

```
5
```

Input example

```
8 10
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 1
2 7
3 6
```

Output example

```
0
```

Language : [C] [C++] [JAVA] [Python]

[HUST] Algorithm Problem Solving
Problem F : Tree(High)

Time Limit: 2000 ms Memory Limit: 64 MB

Problem

The tree T has the structure shown in Figure 1 below, the circle is called 'vertex', and the line connecting vertex and vertex is called 'edge'. In particular, the vertex at the top is called the 'root', and there is only one. N vertices are represented by numbers 1 through N, and the root is always 1.

The vertices v1 and v1 + 1 are connected to the edges by v0 = v, vm = w, with the path connecting the two vertices v and w to the order list (v0, v1, ..., vm). There is only one path in the tree that always connects two vertices between any two vertices v and w. For example, in Figure 1, the only path between vertices 3 and 11 is (3, 4, 1, 7, 11).

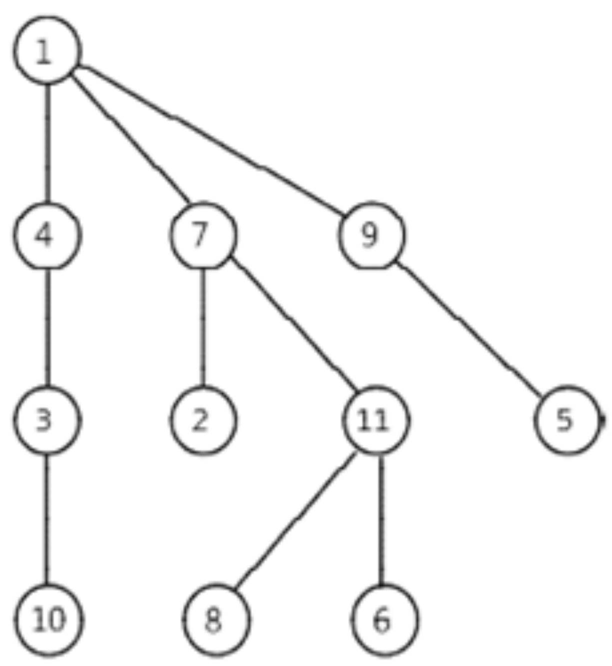


그림 1

In each vertex v, the vertex on P is called the 'parent vertex' of v from the vertex v connected to the root r and the vertex connected to the edge v by the edge. For example, in Figure 1, the parent vertices of 4, 7, and 9 are 1, and the parent vertices of 2 and 11 are 7.

If you remove an edge that connects any two vertices in the tree T, there may be a vertex pair that does not have a path in addition to the two vertices. You must answer a query such as "**Is there a path between vertex v and w**?". For example, if we remove the edge between 7 and 11 in Figure 1, there is no path connecting 8 and 5.

When a tree is given, and edge removal information and queries are given in an arbitrary order, write a program that does the work in order and prints the answer to the query.

Input format

The following information is given as standard input. The first line gives two integers N and Q (1 ≤ N, Q ≤ 200,000) that represent the number of vertices in the tree and the number of queries. The i-th line of the next N-1 lines is given the integer a representing the parent vertex of the vertex i + 1 (1 ≤ a ≤ N). Each of the following Q lines is given three integers b, c, d. If d = 0, it means that only queries that ask whether there is a path connecting b and c are performed. If d = 1, perform a query to see if there is a path connecting b and c, and if the answer is "YES", then remove the edge connecting b's parent vertex and b. If the answer is "NO", remove the edge connecting c's parent vertex and c. If the edge to be removed does not exist originally or is already removed, there is no edge to be removed.

Output format

Output to the standard output in Q lines the answers to the query in order. If there is a path for each line, it outputs YES or NO.

Constraints on Partial Problems • Partial problem 1: 1 ≤ N ≤ 1,000, 1 ≤ Q ≤ 1,000, and the parent vertex of vertex i is the vertex i - 1 (i = 1, ..., N).• Partial problem 2: It corresponds to 13 out of 100 points of the total score, 1 ≤ N ≤ 1,000, 1 ≤ Q ≤ 1,000.• Partial problem 3: It corresponds to 14 out of 100 points of the total score, 1 ≤ N ≤ 3,000, 1 ≤ Q ≤ 200,000.• Partial problem 4: It corresponds to 19 out of 100 points of the total score. The parent vertices of all vertices except root are different from each other.• Partial problem 5: The total score is 45 out of 100 points. There are no constraints other than the original constraint.

Input example

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

Output example

```
YES
YES
NO
NO
```

Input example

```
11 7
7
4
1
9
11
1
11
1
3
7
11 9 1
8 5 0
3 9 0
6 3 1
10 9 1
3 10 1
1 4 1
```

Output example

```
YES
NO
YES
NO
NO
YES
YES
```

Language : [C] [C++] [JAVA] [Python]

[HUST] Algorithm Problem Solving
Problem C : Picking shellfish

Time Limit: 1000 ms Memory Limit: 64 MB

Problem

On the seashore, several areas are divided into square lattices. There is only one household in each area, and the fish market is located in the upper left corner. (One household also lives in the area where the fish market is located.)

To move from each region to the fish market, use only the following two methods.

1. Go to the area directly above

2. Go to the area to the immediate left

People living in each area go to the fish market every day, picking shellfish from the passing area and selling them in the fish market. (Shells can also be picked up in areas with departure and aquatic markets.)

In each area there is a maximum number of shells that one household can pick up to pass through the area for nature protection. (When several households pass by, the shells are enough to get the maximum amount of shellfish for each household.)

For example, suppose that the size of the grid is 3 (rows) × 3 (rows) and the maximum number of shells that one household can pick up per region is the same as the following table.

3	2	7
4	2	6
5	3	8

3	5	12
7	9	18
12	15	26

The maximum number of shellfish that can be sold in the fish market in each region in a day is shown in the table on the right. For example, starting from the bottom right-hand grid and picking up the shells at the maximum, move up two times and move left two times, totaling 8 + 6 + 7 + 2 + 3 = 26 shells. In addition, the total number of shells that can be sold to the fish market in the nine regions of the city is 3 + 5 + 12 + 7 + 9 + 18 + 12 + 15 + 26 = 107.

Sincere officials at the time of settlement periodically examine the number of shells in each area and modify the maximum number of shells a household can pick up. However, sudden changes are dangerous, so it is possible to adjust the maximum value by +1 or -1. The maximum value of the number of clams in uncoordinated areas remains unchanged. For example, in the table on the left above, if the first row of the grid and the second row of the second row are changed to 3, the maximum number of seashells that can be picked in each region and the maximum number of seashells that can be sold to the fisheries market in each region are .

3	**3**	7
4	2	6
5	3	8

3	6	13
7	9	19
12	15	27

Given the initial value of the maximum number of seashells that can be caught per lattice cell and taking the change in the maximum number of seashells that can be picked up in the lattice cell as input, the maximum number of seashells Write a program that computes and prints the sum.

Input format

The following information is given as standard input: In the first row, an integer N (2≤N≤1,500) representing the number of rows (columns) of the grid is given. In the next N lines, the number of shells that can be picked from each grid cell is given one row at a time, starting from the top row. The values of one row are listed one by one from the values in the leftmost column. Given values are between 0 and 1,000. The next N lines are given one change command on each line. The first letter of the change order is U or D. Then, one blank is given and two natural numbers are given, the first being the row number and the second the column number. If the first letter is U, the number of shells that can be picked up in the grid box corresponding to the row number and column number increases by one. D, the number of shells that can be picked in the grid box decreases by one. There is no case where the result of the decrease becomes negative. For each change, you should print the value specified below. Each given change instruction is applied to the result of all previous changes applied.

Output format

Outputs the sum of the maximum number of shells that can be sold in all regions based on the input of each grid cell initially on the standard output. Then, after applying the change command for each change command, it outputs the sum of the maximum number of shells that can be sold in all regions. Note that the total output is reduced by N + 1.

Constraints on Partial Problems

- Partial problem 1: It corresponds to 12 out of 100 points of the total score and N ≤100.
- Partial problem 2: The total score is 34 out of 100 points. All changes are D and only the input with the first line of output less than 20,000,000 is given.
- Partial problem 3: The total score is 54 out of 100 points. There are no constraints other than the original constraint.

Input example

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

Output example

```
107
111
110
114
```

Language : [C] [C++] [JAVA] [Python]