A - Adjacent Delete

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 700 points

Problem Statement

You are given a length-N sequence $A = (A_1, A_2, \ldots, A_N)$.

You will repeatedly perform the following operation until the sequence has length at most 1: choose two adjacent numbers and remove both from the sequence.

The score obtained in one operation is the absolute difference of the two chosen numbers.

Find the maximum possible total score obtained.

Constraints

- $2 \leq N \leq 3 \times 10^5$
- $1 \leq A_i \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1 \quad A_2 \quad \ldots \quad A_N
\end{array}
$$

Output

Print the maximum possible total score obtained.

Sample Input 1

```
4
1 2 5 3
```

Sample Output 1

```
5
```

First, remove A_2 and A_3. The score obtained is $|A_2 - A_3| = 3$.

Next, remove A_1 and A_4. Note that, because of the previous operation, these two numbers are now adjacent. The score obtained is $|A_1 - A_4| = 2$.

Hence, the total score obtained is 5.

It is impossible to achieve a total score of 6 or greater, so print 5.

Sample Input 2

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

Sample Output 2

```
14
```

Sample Input 3

```
5
1 1 1 1 1
```

Sample Output 3

```
0
```

B - Torus Loop

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 700 points

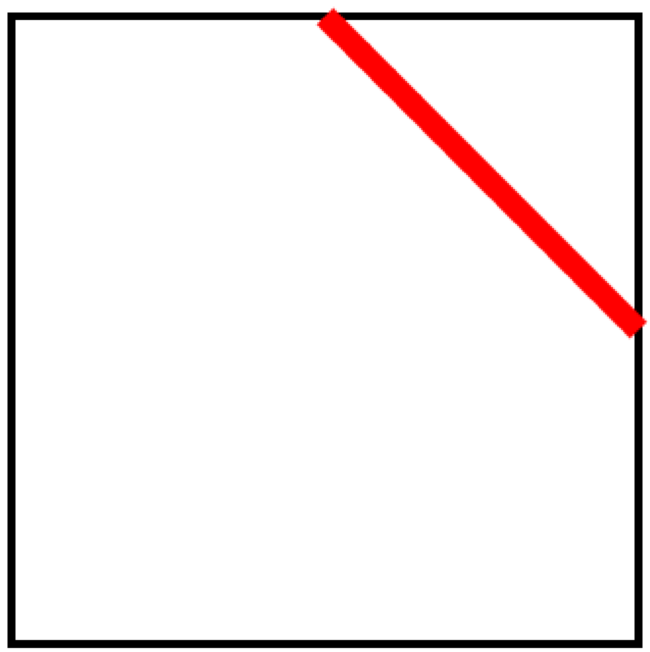
Problem Statement

There is a grid of H rows and W columns. The rows are numbered $0, 1, \ldots, H-1$ from top to bottom, and the columns are numbered $0, 1, \ldots, W-1$ from left to right. Let (i, j) denote the cell at row i and column j.

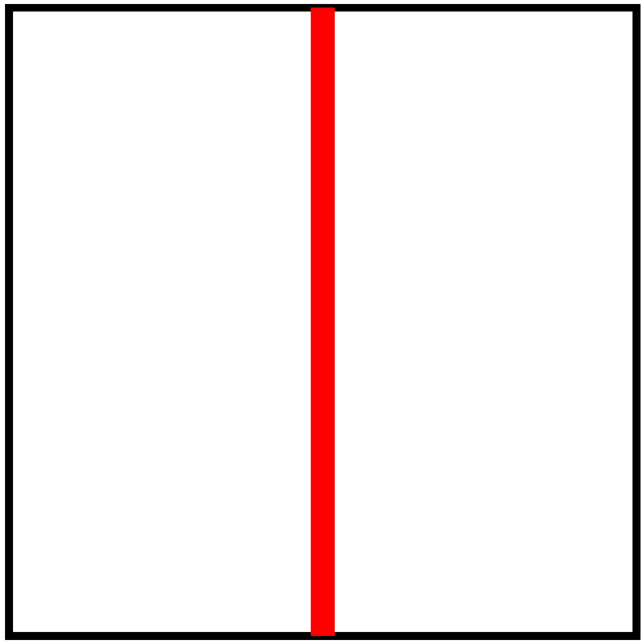
You are given H strings $S_0, S_1, \ldots, S_{H-1}$, each of which is of length W and consists of A and B.

In each cell, one of the following two types of tiles is placed. Let S_{ij} denote the $(j+1)$-th character $(0 \leq j \leq W-1)$ of the string S_i. The type of tile placed in cell (i, j) is S_{ij}.

- Type A: A single line segment is drawn on the tile's surface, connecting the midpoints of two adjacent edges.



- Type B: A single line segment is drawn on the tile's surface, connecting the midpoints of two opposite edges.



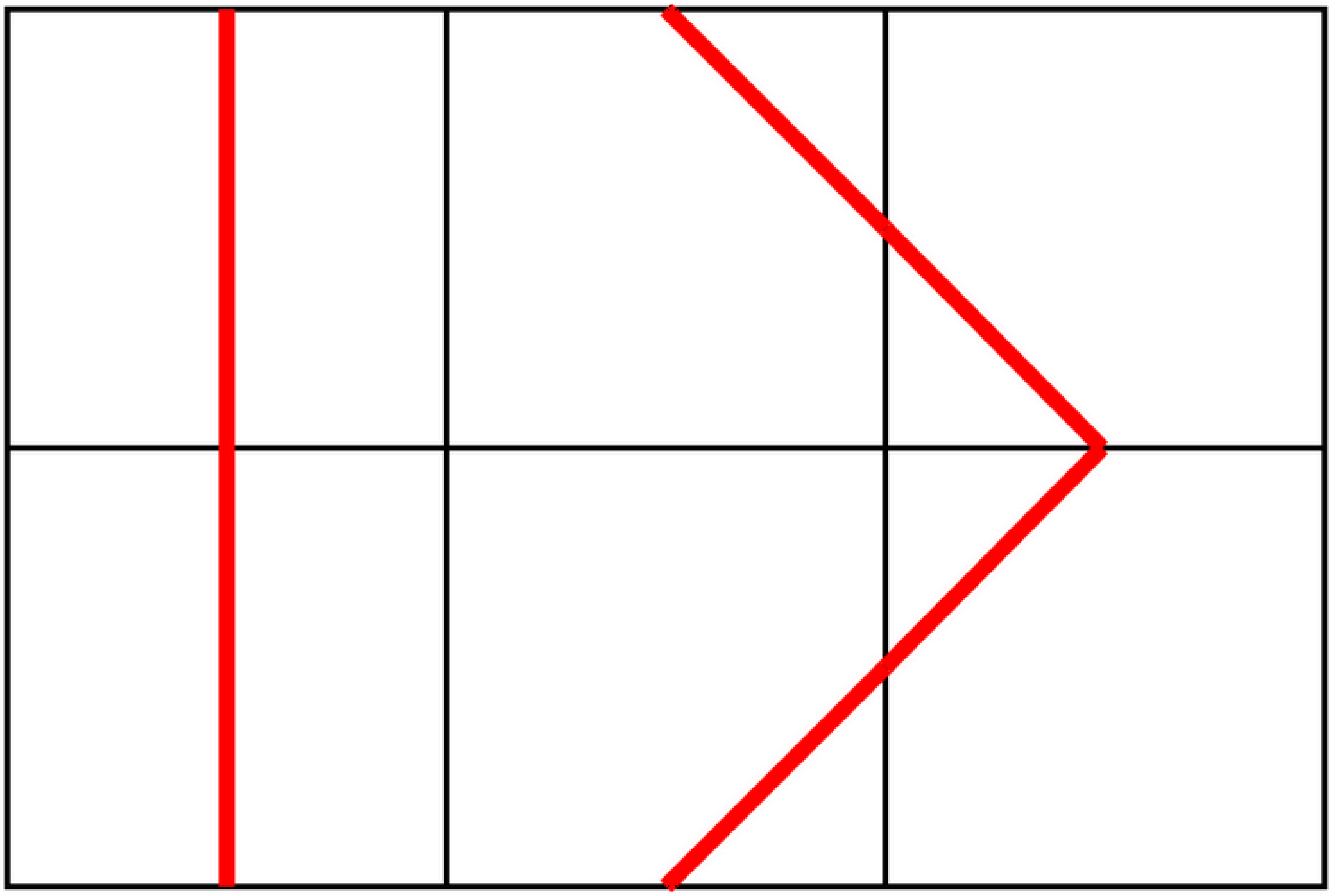
These tiles can be freely rotated. When focusing only on the pattern formed by the line segments, there are four ways to rotate a Type-A tile and two ways to rotate a Type-B tile. Therefore, if we distinguish placements only by the pattern of line segments, the number of ways to place the tiles is $4^a \times 2^b$, where a is the number of Type-A tiles and b is the number of Type-B tiles.

Among these ways, print the number, modulo 998244353, of ways such that the line segments on the tiles have no dead ends when viewing the grid as a torus.

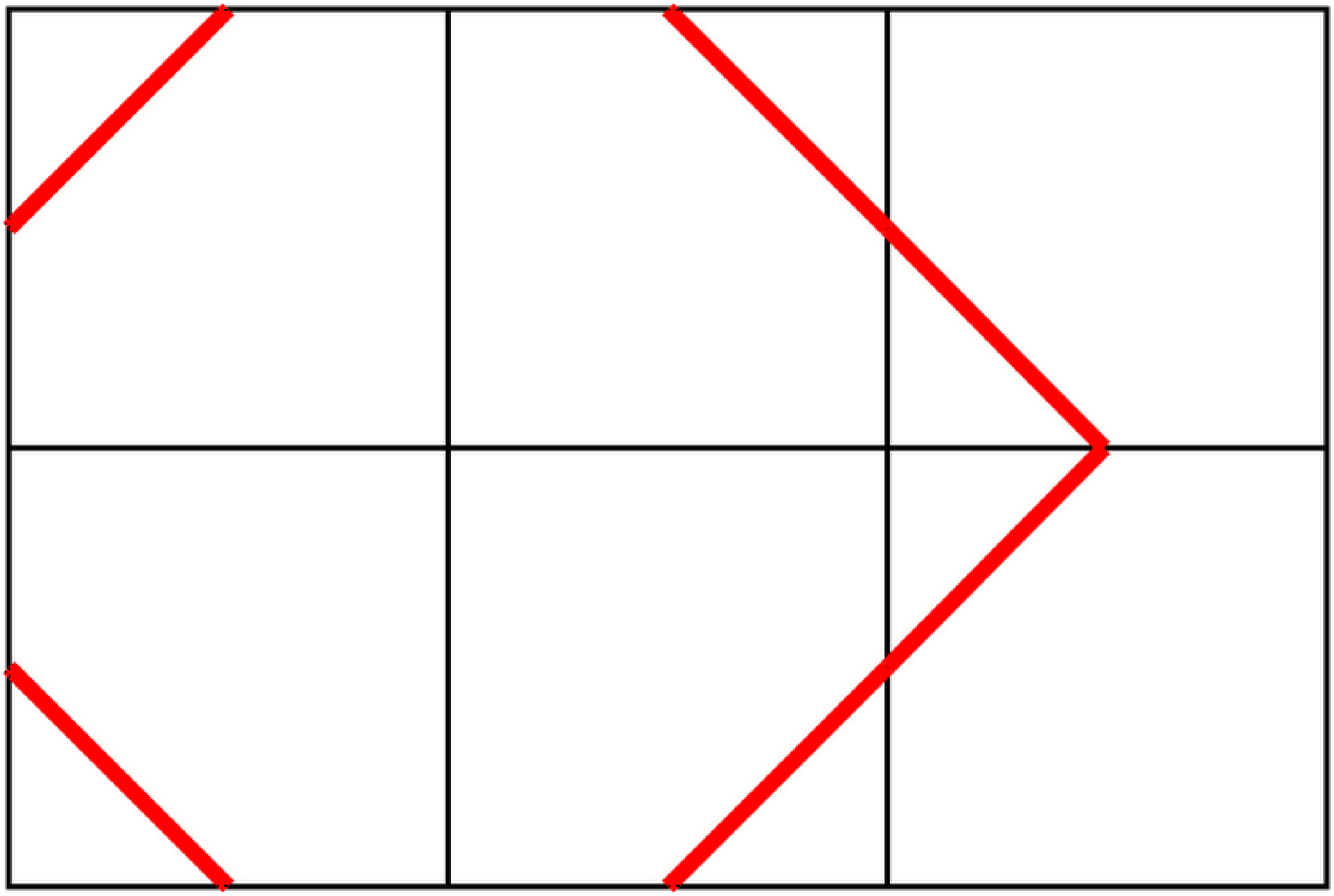
Here, "the line segments on the tiles have no dead ends when viewing the grid as a torus" if and only if the following two conditions are satisfied for every cell (i, j):

- Both of the following exist, or neither of the following exists:
  - the line segment drawn in the cell (i, j), whose endpoint is the midpoint of the right edge of the cell (i, j)
  - the line segment drawn in the cell $(i, (j + 1) \bmod W)$, whose endpoint is the midpoint of the left edge of the cell $(i, (j + 1) \bmod W)$
- Both of the following exist, or neither of the following exists:
  - the line segment drawn in the cell (i, j), whose endpoint is the midpoint of the bottom edge of the cell (i, j)
  - the line segment drawn in the cell $((i + 1) \bmod H, j)$, whose endpoint is the midpoint of the top edge of the cell $((i + 1) \bmod H, j)$

For example, the following placement satisfies the condition:



The following placement does not satisfy the condition. Specifically, while there is no line segment whose endpoint is the midpoint of the right edge of the tile in cell $(0, 2)$, there is a line segment whose endpoint is the midpoint of the left edge of the tile in cell $(0, 0)$, so the condition is not satisfied.



You are given T test cases; solve each of them.

Constraints

- $1 \leq T \leq 10^5$
- $2 \leq H, W$
- $HW \leq 10^6$
- $S_i \, (0 \leq i \leq H-1)$ are length-W strings consisting of A and B.
- The sum of HW over all test cases is at most 10^6.
- T, H, and W are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{aligned}
&T \\
&case_1 \\
&case_2 \\
&\vdots \\
&case_T
\end{aligned}
$$

Each case is given in the following format:

$$
\begin{aligned}
&H \quad W \\
&S_0 \\
&S_1 \\
&\vdots \\
&S_{H-1}
\end{aligned}
$$

Output

For each test case, print the number, modulo 998244353, of placements that satisfies the condition, in separate lines.

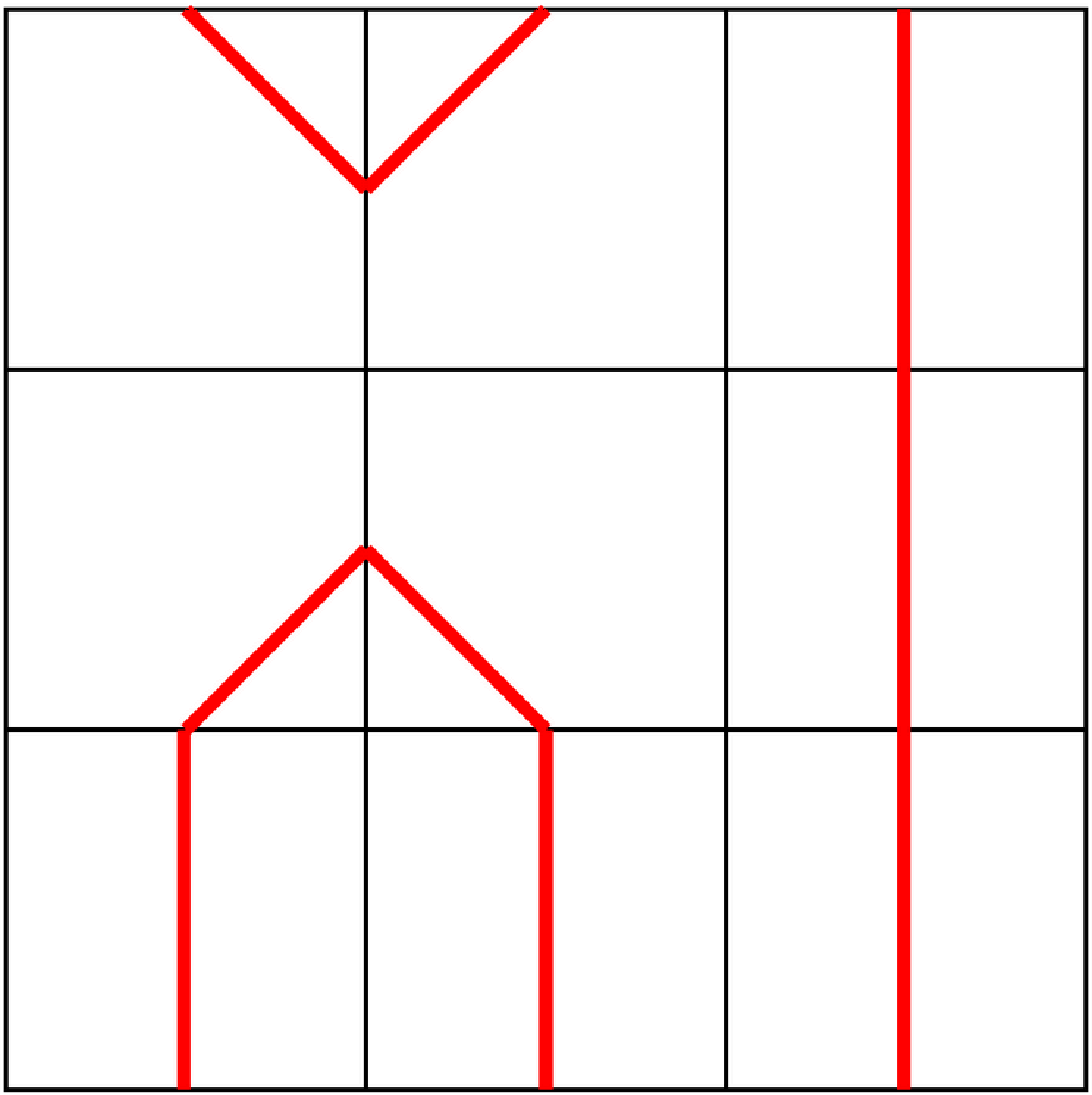
Sample Input 1

```
3
3 3
AAB
AAB
BBB
3 3
BBA
ABA
AAB
3 4
BAAB
BABA
BBAA
```

Sample Output 1

```
2
0
2
```

One valid placement for the first test case is shown in the following image:



C - Strongly Connected

Time Limit: 4 sec / Memory Limit: 1024 MiB

Score : 900 points

Problem Statement

There is a directed graph with $2N$ vertices and $2N-1$ edges. The vertices are numbered $1, 2, \ldots, 2N$, and the i-th edge is a directed edge from vertex i to vertex $i+1$.

You are given a length-$2N$ string $S = S_1 S_2 \ldots S_{2N}$ consisting of N `W`s and N `B`s. Vertex i is colored white if S_i is `W`, and black if S_i is `B`.

You will perform the following series of operations:

- Partition the $2N$ vertices into N pairs, each consisting of one white vertex and one black vertex.
- For each pair, add a directed edge from the white vertex to the black vertex.

Print the number, modulo 998244353, of ways to partition the vertices into N pairs such that the final graph is strongly connected.

▶ Notes on strongly connectedness

Constraints

- $1 \leq N \leq 2 \times 10^5$
- S is a length $2N$ string consisting of N `W`s and N `B`s.
- N is an integer.

Input

The input is given from Standard Input in the following format:

```
N
S
```

Output

Print the number, modulo 998244353, of ways to partition the vertices into N pairs so that the final graph is strongly connected.

Sample Input 1

```
2
BWBW
```

Sample Output 1

```
1
```

Vertices $2, 4$ are white, and vertices $1, 3$ are black.

Let (u, v) denote an edge from vertex u to vertex v.

If we pair up vertices as $(2, 1), (4, 3)$, the final graph have the edges $(1, 2), (2, 3), (3, 4), (2, 1), (4, 3)$. In this case, for example, it is impossible to travel from vertex 3 to vertex 1 by following edges, so this graph is not strongly connected.

If we pair up vertices as $(2, 3), (4, 1)$, the final graph have the edges $(1, 2), (2, 3), (3, 4), (2, 3), (4, 1)$. This graph is strongly connected.

Therefore, there is exactly 1 way to pair up the vertices that satisfies the condition.

Sample Input 2

```
4
BWWBWBWB
```

Sample Output 2

```
0
```

No matter how you pair up the vertices, you cannot satisfy the condition.

Sample Input 3

```
9
BWWBWBBBWWBWBBWWBW
```

Sample Output 3

```
240792
```

D - Roadway

Time Limit: 3 sec / Memory Limit: 1024 MiB

Score : 900 points

Problem Statement

There are N towns, numbered $1, 2, \ldots, N$, arranged in a line in this order.

There are $N-1$ roads connecting adjacent towns: road j $(1 \leq j \leq N-1)$ connects towns j and $j+1$. For each road j, you can set a **strength** w_j (an integer that may be negative).

When a person travels along a road, their **stamina** changes. Specifically, if a person with stamina x travels along road j, their stamina becomes $x + w_j$.

There are M people who will now move between these towns.

Person i $(1 \leq i \leq M)$ starts with stamina 0 at town S_i and travels to town T_i via the shortest path. It is guaranteed that $|S_i - T_i| > 1$. Also, $(S_i, T_i) \neq (S_j, T_j)$ if $i \neq j$.

Person i's requirement is as follows:

> When departing Town S_i and when arriving at Town T_i, their stamina should be exactly 0. At every other town, their stamina should always be a positive integer.

Assume that there are no changes to stamina other than those due to traveling along roads as described above.

Process Q queries. For the k-th query $(1 \leq k \leq Q)$, if it is possible to set the strengths of the roads so that the requirements of all people $L_k, L_k + 1, \ldots, R_k$ are satisfied, print `Yes`; otherwise, print `No`.

Constraints

- $3 \leq N \leq 4 \times 10^5$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq S_i, T_i \leq N$
- $|S_i - T_i| > 1$
- $(S_i, T_i) \neq (S_j, T_j)$ $(i \neq j)$
- $1 \leq L_k \leq R_k \leq M$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \quad Q \\
S_1 \quad T_1 \\
S_2 \quad T_2 \\
\vdots \\
S_M \quad T_M \\
L_1 \quad R_1 \\
L_2 \quad R_2 \\
\vdots \\
L_Q \quad R_Q
\end{array}
$$

Output

Print Q lines.

The k-th line should contain `Yes` if there is a way to set the strengths of the roads so that the requirements of all people $L_k, L_k + 1, \ldots, R_k$ are satisfied, and `No` otherwise.

Sample Input 1

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

Sample Output 1

```
Yes
No
```

For the first query, consider setting the strengths of roads $1, 2, 3, 4$ to $1, -1, 1, -1$, respectively.

- Person 1 starts at town 4 with stamina 0, visits town 3 with stamina 1, and arrives at town 2 with stamina 0.
- Person 2 starts at town 1 with stamina 0, visits town 2 with stamina 1, and arrives at town 3 with stamina 0.
- Person 3 starts at town 3 with stamina 0, visits town 4 with stamina 1, and arrives at town 5 with stamina 0.

Thus, this configuration satisfies the requirements of persons $1, 2, 3$, so print `Yes` on the first line.

For the second query, it is impossible to satisfy the requirements of persons $2, 3, 4$ simultaneously, so print `No`.

Sample Input 2

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

Sample Output 2

```
No
Yes
Yes
```