E. Quick Tortoise

John Doe has a field, which is a rectangular table of size $n \times m$. We assume that the field rows are numbered from 1 to n from top to bottom, and the field columns are numbered from 1 to m from left to right. Then the cell of the field at the intersection of the x-th row and the y-th column has coordinates $(x; y)$.

We know that some cells of John's field are painted white, and some are painted black. Also, John has a tortoise, which can move along the white cells of the field. The tortoise can get from a white cell with coordinates $(x; y)$ into cell $(x + 1; y)$ or $(x; y + 1)$, if the corresponding cell is painted white. In other words, the turtle can move only along the white cells of the field to the right or down. The turtle can not go out of the bounds of the field.

In addition, John has q queries, each of them is characterized by four numbers x_1, y_1, x_2, y_2 ($x_1 \leq x_2, y_1 \leq y_2$). For each query John wants to know whether the tortoise can start from the point with coordinates $(x_1; y_1)$, and reach the point with coordinates $(x_2; y_2)$, moving only along the white squares of the field.

Input

The first line contains two space-separated integers n and m ($1 \leq n, m \leq 500$) — the field sizes.

Each of the next n lines contains m characters "#" and ".": the j-th character of the i-th line equals "#", if the cell $(i; j)$ is painted black and ".", if it is painted white.

The next line contains integer q ($1 \leq q \leq 6 \cdot 10^5$) — the number of queries. Next q lines contain four space-separated integers x_1, y_1, x_2 and y_2 ($1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq m$) — the coordinates of the starting and the finishing cells. It is guaranteed that cells $(x_1; y_1)$ and $(x_2; y_2)$ are white.

Output

For each of q queries print on a single line "Yes", if there is a way from cell $(x_1; y_1)$ to cell $(x_2; y_2)$, that meets the requirements, and "No" otherwise. Print the answers to the queries in the order, in which the queries are given in the input.

Examples

input

```
3 3
...
.##
.#.
5
1 1 3 3
1 1 1 3
1 1 3 1
1 1 1 2
1 1 2 1
```

output

```
No
Yes
Yes
Yes
Yes
```

input

```
5 5
.....
```

```
.###.
.....
.###.
.....
5
1 1 5 5
1 1 1 5
1 1 3 4
2 1 2 5
1 1 2 5
```

output

```
Yes
Yes
Yes
No
Yes
```