
Problem A. Escape

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Given a maze of size $n \times m$. The rows are numbered $1, 2, \dots, n$ from top to bottom while the columns are numbered $1, 2, \dots, m$ from left to right, which means that $(1, 1)$ is the top-left corner and that (n, m) is the bottom-right corner. And for each cell of size 1×1 , it is either blank or blocked.

There are a robots above the maze. For i -th robot, it is initially positioned exactly above the cell $(1, p_i)$, which can be described as $(0, p_i)$. And the initial moving direction of the robots are all downward, which can be written as $(1, 0)$ in the vector form.

Also, there are b exits below the maze. For i -th exit, it is positioned exactly below the cell (n, e_i) , which can be described as $(n + 1, e_i)$.

Now, you want to let the robots escape from the maze by reaching one of the exits. However, the robots are only able to go straight along their moving directions and can't make a turn. So you should set some turning devices on some blank cells in the maze to help the robots make turns. There are 4 types of turning devices:

- “NE-devices”: make the robots coming from above go rightward, and make the robots coming from right go upward. Coming from left or below is illegal.
- “NW-devices”: make the robots coming from above go leftward, and make the robots coming from left go upward. Coming from right or below is illegal.
- “SE-devices”: make the robots coming from below go rightward, and make the robots coming from right go downward. Coming from left or above is illegal.
- “SW-devices”: make the robots coming from below go leftward, and make the robots coming from left go downward. Coming from right or above is illegal.

For each cell, the number of turning devices on it can not exceed 1. And collisions between the robots are ignored, which allows multiple robots to visit one same cell even at the same time.

You want to know if there exists some schemes to set turning devices so that all the a robots can reach one of the b exits after making a finite number of moves without passing a blocked cell or passing a turning device illegally or going out of boundary(except the initial position and the exit). If the answer is yes, print “Yes” in a single line, or print “No” if the answer is no.

Input

The first line contains one positive integer T ($1 \leq T \leq 10$), denoting the number of test cases.

For each test case:

The first line contains four positive integers n, m, a, b ($1 \leq n, m \leq 100, 1 \leq a, b \leq m$), denoting the number of rows and the number of columns in the maze, the number of robots and the number of exits respectively.

Next n lines each contains a string of length m containing only “0” or “1”, denoting the initial maze, where cell (i, j) is blank if the j -th character in i -th string is “0”, while cell (i, j) is blocked if the j -th character in i -th string is “1”.

The next line contains a integers p_i ($1 \leq p_i \leq m$), denoting the initial positions $(0, p_i)$ of the robots.

The next line contains b integers e_i ($1 \leq e_i \leq m$), denoting the positions $(n + 1, e_i)$ of the exits.

It is guaranteed that all p_i s are pairwise distinct and that all e_i s are also pairwise distinct.

Output

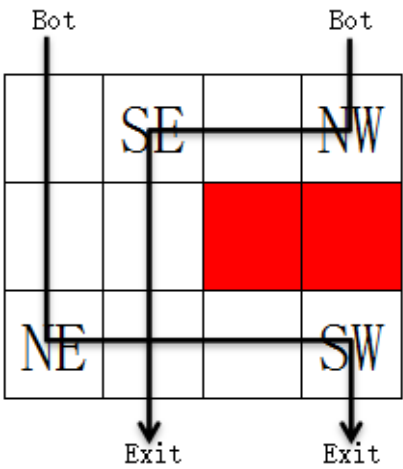
Output T lines each contains a string “Yes” or “No”, denoting the answer to corresponding test case.

Example

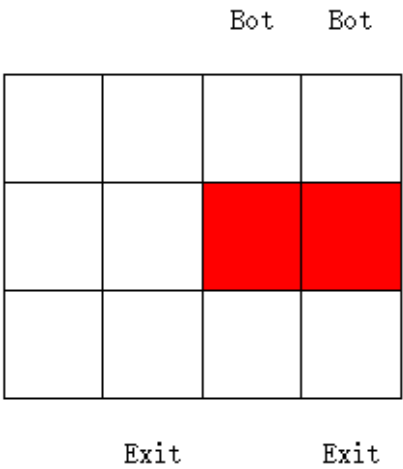
standard input	standard output
2	Yes
3 4 2 2	No
0000	
0011	
0000	
1 4	
2 4	
3 4 2 2	
0000	
0011	
0000	
3 4	
2 4	

Note

Please look at the following illustration.



Case 1



Case 2