Find a Way

Run-time Limit: 3 seconds
Memory Limit: 64 MB

DESCRIPTION

Bagong was dropped in the middle of forest, and from here on, he should find a way out. He knows where he should go, but the forest has many obstacles. He has many ways to remove those obstacles, but he wants to minimize the number of obstacles he meets.

Help him find the minimum obstacles he needs to remove so that he can go to the destination.

INPUT FORMAT

The first line of input contains an integer T ($1 \leq T \leq 1000$) denoting the number of cases. For each case, it starts with the dimension of the forest n x m ($1 \leq n, m \leq 200$). Each n lines contains m characters consists of '.' and '#', '.' means the specified block has no obstacle, and '#' means the specified block has an obstacle. The next following line contain starting coordinates of Rudi (xs,ys) and the next line contains his destination (xf,yf). It's guaranteed both coordinates lies within the forest.

OUTPUT FORMAT

For each case, output "Case #X: Y", where X is the case number starts from 1, and Y the minimum number of obstacles he has to remove.

INPUT EXAMPLE

```
2
5 5
.....
.....
.....
.....
.....
1 1
4 4
5 10
##########
#.#.....##
#####.####
#.######..
#.....####
1 1
4 9
```

OUTPUT EXAMPLE

```
Case #1: 0
Case #2: 4
```