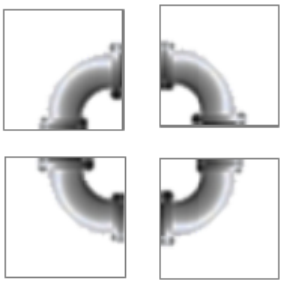
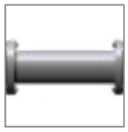
7844 Ultimate Pipe Game

Pipe games are interesting and hard puzzle games. In these games, our mission is to connect the pipes to make the water flow in the pipeline without leaking out from a source to a destination. Today we are playing a new generation of pipe game, the Ultimate Pipe Game.

In this game, we have a grid of m rows and n columns. Cells on the grid are either empty or blocked. The cell at row i and column j is denoted as cell (i, j). We only put pipes on empty cells and each cell can contain only one pipe. There are 3 kinds of pipes: curved pipes, horizontal pipes and vertical pipes. Note that a curved pipe can be rotated as shown in the picture below, but we cannot rotate a horizontal pipe to make a vertical pipe or vice versa.



Curved pipe which can be rotated



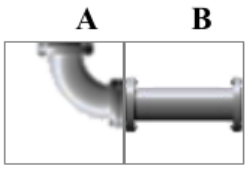
Horizontal pipe

Vertical pipe

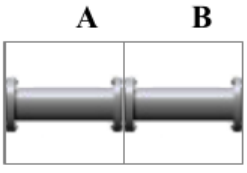
By putting pipes on empty cells, water from a cell can travel to its adjacent cell if and only if the pipes in two cells are connected by their heads.

Our mission in this Ultimate Pipe Game is to place pipes on the empty cells with following requirements:

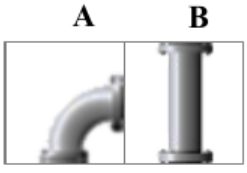
- We need to put pipes on **every** empty cell and each empty cell must contain exactly one pipe.
- We have to make sure water in the pipeline will not leak out. Starting from any empty cell with a pipe, assuming that we have water flows out from that pipe in one of the two directions, water can travel through the pipeline to other cells without leaking out and return to the starting cell. In this case, we call it a **cycle pipeline**.
- We can have multiple disjoint **cycle pipelines**.



A and B are connected

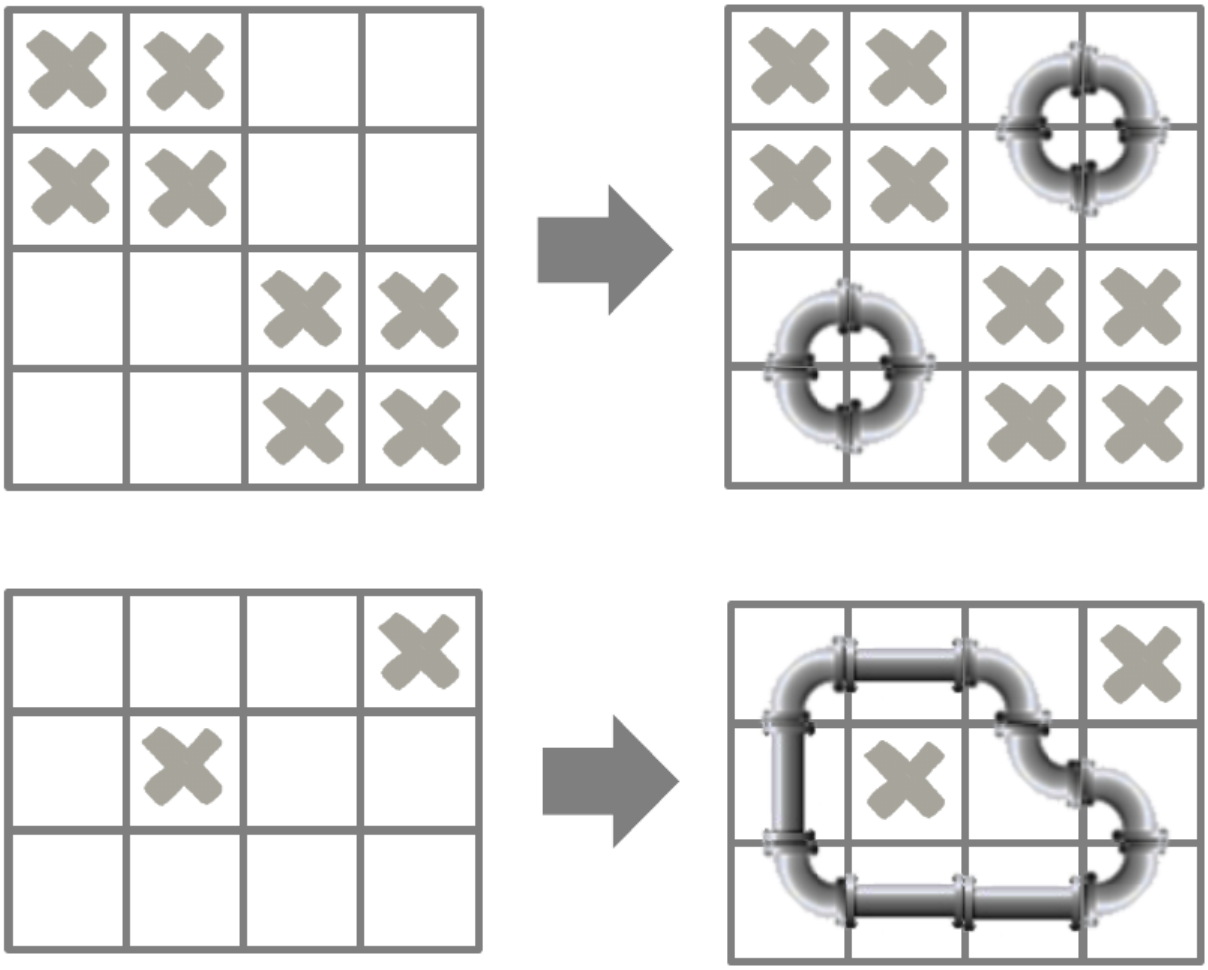


A and B are connected



A and B are not connected

- Putting a horizontal pipe and a vertical pipe at cell (i, j) costs $h_{i,j}$ and $v_{i,j}$ coins, respectively. Putting in a curved pipe is free.



Your task is to find a way, if exists, to put pipes on empty cells to minimize the total number of coins. You can assume that there are unlimited number of pipes.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 100. The following lines describe the datasets.

Each dataset is described by the following lines:

- First line contains two integers m and n ($2 \leq m, n \leq 20$).
- The next m lines describe the grid where each line contains n characters. The j-th character at i-th line denotes the state of cell (i, j): '`.`' for an empty cell or '`#`' for a blocked cell.
- In the next m lines, each line contains n integers denoting the cost of putting a horizontal pipe on a cell. The j-th number on the i-th line is an integer $h_{i,j}$ ($0 \leq h_{i,j} \leq 100$) where $h_{i,j}$ is 0 for a blocked cell.
- In the last m lines, each line contains n integers denoting the cost of putting a vertical pipe on a cell. The j-th number on the i-th line is an integer $v_{i,j}$ ($0 \leq v_{i,j} \leq 100$) where $v_{i,j}$ is 0 for a blocked cell.

Output

For each dataset, write out the result in the following format.

- If you can find a way to put pipes on empty cells to fulfill the requirements, write out on one line the string '`YES` y' where y is the minimum total number of coins we need to pay.
- Otherwise, write out on one line the string '`NO`'.

Sample Input

```
3
4 4
##..
##..
..##
..##
0 0 1 2
0 0 3 0
1 2 0 0
2 3 0 0
0 0 1 2
0 0 3 0
1 2 0 0
2 3 0 0
3 4
...#
.#..
....
1 2 3 0
4 0 1 2
3 1 2 3
3 2 1 0
5 0 2 2
3 1 2 3
3 3
...
...
...
0 0 0
0 0 0
0 0 0
1 1 1
1 1 1
1 1 1
```

Sample Output

```
YES 0
YES 10
NO
```