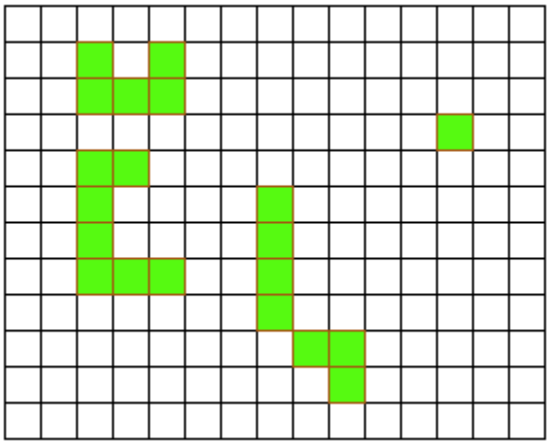
# Connect the Islands

**Note:** This is a problem taken from a 2016 ACM-ICPC local contest.

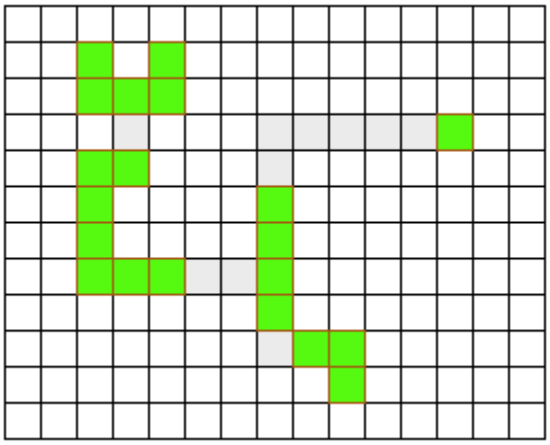
You are given a rectangular map of *RxC* cells showing a number of small islands surrounded by sea. Each cell contains either land or sea water (but not both). Two cells are considered connected if they share an edge or you may travel between those two cells using other connected cells. An island is a maximal group of land cells that are connected. For instance, the following 12x15 map has 5 islands (painted in green):



To facilitate inter-travel, the governments of these island nations wish to build some number of bridges to connect the islands. Each bridge connects exactly two islands. The cost of building a bridge is equal to the number of cells that it occupies on the map. As all bridges have distinct heights, they do not intersect each other.

What is the minimum cost to build the bridges required to connect all the islands?

For the example above, here is a way to build bridges (denoted by the grey squares) that minimizes the total cost:



**Input specification**

In the first line of input, we have a single integer $\boldsymbol{T}$ **(1 ≤ $\boldsymbol{T}$ ≤ 10)**, which corresponds to the number of test cases.

Each test case consists of two lines:

1. Line 1 contains exactly two space-separated integers $\boldsymbol{R}$ **(3 ≤ $\boldsymbol{R}$ ≤ 50)**, $\boldsymbol{C}$ **(3 ≤ $\boldsymbol{C}$ ≤ 50)**.
2. Lines 2 through $R$+1 contain $C$ characters each, where the *jth* character in the *ith* row corresponds to cell ($i$, $j$) in the map. Each character is either the '.' or 'X', representing sea and land respectively. It is guaranteed that cells in the border of the map (ie, in the first row, last row, first column or last column) contain water.

For each test case, the number of islands is at least 2 and at most 300.

**Output specification**

For each test case, in the order given in the input, print the minimum cost to build bridges that connect all islands.

**Sample input**

```
2
12 15
...............
..X.X..........
..XXX..........
............X..
..XX...........
..X....X.......
..X....X.......
..XXX..X.......
.......X.......
........XX.....
.........X.....
...............
5 5
.....
.X...
...X.
.X...
.....
```

**Sample output**

```
10
3
```