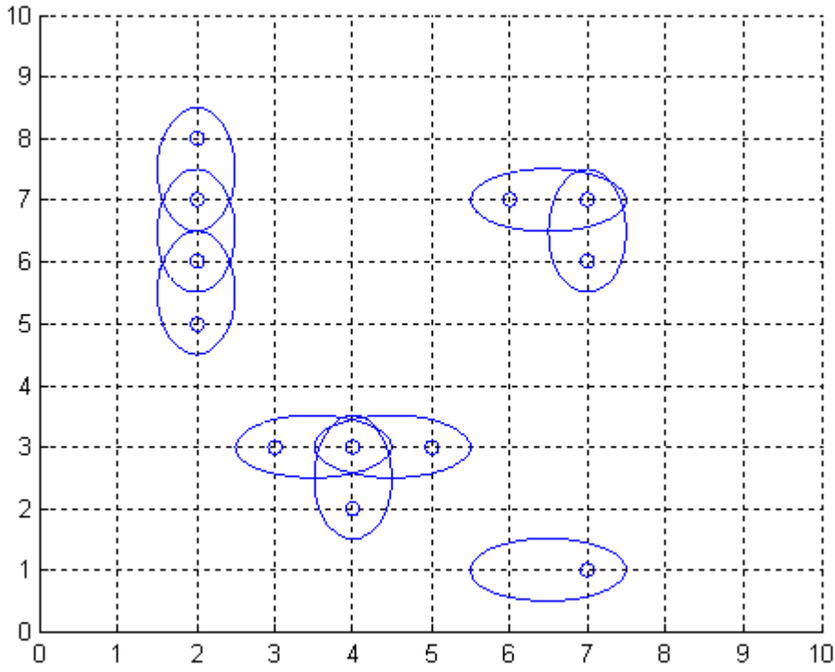
The Global Aerial Research Centre has been allotted the task of building the fifth generation of mobile phone nets in Sweden. The most striking reason why they got the job, is their discovery of a new, highly noise resistant, antenna. It is called 4DAir, and comes in four types. Each type can only transmit and receive signals in a direction aligned with a (slightly skewed) latitudinal and longitudinal grid, because of the interacting electromagnetic field of the earth. The four types correspond to antennas operating in the directions north, west, south, and east, respectively. Below is an example picture of places of interest, depicted by twelve small rings, and nine 4DAir antennas depicted by ellipses covering them.



Obviously, it is desirable to use as few antennas as possible, but still provide coverage for each place of interest. We model the problem as follows: Let A be a rectangular matrix describing the surface of Sweden, where an entry of A either is a point of interest, which must be covered by at least one antenna, or empty space. Antennas can only be positioned at an entry in A. When an antenna is placed at row r and column c, this entry is considered covered, but also one of the neighbouring entries $(c+1, r)$, $(c, r+1)$,$(c-1, r)$, or $(c, r-1)$, is covered depending on the type chosen for this particular antenna. What is the least number of antennas for which there exists a placement in A such that all points of interest are covered?

Input

On the first row of input is a single positive integer n, specifying the number of scenarios that follow. Each scenario begins with a row containing two positive integers h and w, with $1 < h < 40$ and $0 < w < 10$. Thereafter is a matrix presented, describing the points of interest in Sweden in the form of h lines, each containing w characters from the set [‘*’, ‘o’]. A ‘*’-character symbolises a point of interest, whereas a ‘o’-character represents open space.

Output

For each scenario, output the minimum number of antennas necessary to cover all ‘*’-entries in the scenario’s matrix, on a row of its own.

Sample Input

```
2
7 9
ooo**oooo
**oo*ooo*
o*oo**o**
ooooooooo
*******oo
o*o*oo*oo
*******oo
10 1
*
*
*
o
*
*
*
*
*
*
```

Sample Output

```
17
5
```