Code Jam 2018 - World Finals

Jurisdiction Restrictions

Problem

The city of Gridtopia is a matrix of square cells ("blocks") with **R** rows and **C** columns; rows are numbered (starting from 1) from top to bottom, and columns are numbered (starting from 1) from left to right. The city is served by **S** different police stations; the i-th station is in the block located in the $\mathbf{R_i}$th row and the $\mathbf{C_i}$th column, and no block contains more than one station.

Each station is only able to patrol blocks that are no more than $\mathbf{D_i}$ blocks away from that station, either horizontally or vertically. That is, the i-th station can only patrol the block in row R' and column C' if $\max(|R' - \mathbf{R_i}|, |C' - \mathbf{C_i}|) \leq \mathbf{D_i}$. Put another way, the i-th station can patrol only blocks within the square of side length $2\mathbf{D_i} + 1$ centered on that station.

As the new police commissioner, you need to assign some blocks within the city to exactly one station that is able to patrol it. Blocks that contain stations and blocks that no station is able to patrol should not be assigned. All other blocks have to be assigned. Moreover, you must distribute this assignment load as evenly as possible among stations. Let A_i denote the number of blocks assigned to the i-th station; then your goal is to minimize the difference between the maximum of all the A_i values and the minimum of all of the A_i values. If you make optimal assignments, what is the smallest possible difference?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each case begins with one line with three integers **R**, **C**, and **S**: respectively, the numbers of rows and columns in the grid of cells, and the number of stations. Then, there are **S** more lines. The i-th of these has three integers $\mathbf{R_i}$, $\mathbf{C_i}$, and $\mathbf{D_i}$: respectively, the row and column in which the i-th station is located, and the parameter that determines which blocks that station is able to patrol, as described above.

Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the difference described above.

Limits

$1 \leq \mathbf{T} \leq 100$.
$2 \leq \mathbf{S} \leq 15$.
$1 \leq \mathbf{R_i} \leq \mathbf{R}$, for all i.
$1 \leq \mathbf{C_i} \leq \mathbf{C}$, for all i.
For all $i \neq j$, $\mathbf{R_i} \neq \mathbf{R_j}$ and/or $\mathbf{C_i} \neq \mathbf{C_j}$. (No two stations are in the same block.)
$1 \leq \mathbf{D_i} < \max(\mathbf{R}, \mathbf{C})$, for all i.
Time limit: 30 seconds per test set.
Memory limit: 1GB.

Test set 1 (Visible)

1 ≤ **R** ≤ 20.
1 ≤ **C** ≤ 20.

Test set 2 (Hidden)

1 ≤ **R** ≤ 10^9.
1 ≤ **C** ≤ 10^9.

Sample

Sample Input

```
2
3 4 2
1 1 1
3 3 2
5 5 2
4 1 2
3 2 2
```

Sample Output

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

In Sample Case #1, the city consists of a grid with 3 rows and 4 columns, with one station in the upper left block and one station in the block to the left of the lower right block. The first station can only patrol the three blocks that touch the edge or corner of its block; every other block is at a horizontal or vertical distance of more than 1 away. The second station can patrol any block in the grid (except for the blocks containing the stations). The difference in number of blocks assigned is minimized if you assign station 1 all three of the blocks it can patrol, and then assign the remaining seven blocks to station 2.

In Sample Case #2, one optimal strategy is to assign the blocks as follows. In this picture, `1` represents station 1, `2` represents station 2, `!` represents a block assigned to station 1, `@` represents a block assigned to station 2, and `.` represents a block assigned to neither station (because neither station can patrol it). Notice that a station's assigned blocks do not need to form a single continuous area.

```
@@@@.
!!!@.
!2!@.
1!!@.
!@!@.
```