# Problem 2 - Mouse Precision Training (`grid`)

To maintain top performance in every game, you should have perfect control of your mouse. To improve your mouse skills you have to practise with a mouse accuracy game.

In this game, there's a grid of $N \times N$ squares. One of these $M$ squares contain a target. The remaining ones are empty.

Each target square is associated to a natural number $i$ representing the maximum number of visits possible for that square. Each time the square is visited $i$ decreases by 1, when $i = 0$ the square can no longer be visited.

Starting from an initial target square, you should move the mouse to the nearest target square, visit it (decreasing its value $i$ by 1) and then continue from that square, searching for the next nearest *different* target square from the last one. When you reach the last target square and all other $i = 0$, you're finished. The goal is to return the total distance travelled by the mouse.

**Parameters and rules:**

1. There are $N^2$ squares on an $NxN$ grid.
2. Each square is identified by its coordinate $(x, y)$. The square on the top left has coordinates $(0, 0)$, the one on bottom right has coordinates $(N - 1, N - 1)$.
3. $M$ target squares, of which the initial target square is the first one in the input file.
4. $i$, the maximum number of visits for each square (each time you visit a square its $i$ decreases by 1. This also applies also when starting from the initial square).
5. The distance between two squares is computed as L1 distance, i.e. the distance between $p$ and $q$ is $d(p, q) = |p_x - q_x| + |p_y - q_y|$.
6. If there are multiple squares with the same distance, always select the nearest one with lower $x$ coordinate. If this isn't possible, select the one with a lower $y$ coordinate.
7. You cannot return to the last target square you visited, except when there are only two target squares on the grid (the current one and the last one).

## Input data

In each file, the first line contains T the number of test cases. Then, for each test case there is:

- a line with two space-separated integers **N** and **M**, the width of the square grid and the number of target squares.
- a line containing the coordinates of the starting target square followed by its maximum number of visits **i**.
- the remaining **M-1** lines are the coordinates of other target squares followed by their maximum number of visits **i**.

## Output data

The output file must contain T lines. For each test case in the input file, the output file must contain a line with the characters:

Case #t: $S$

where t is the test case number, from 1 to T, and $S$ is the solution of the test case: the sum of all the travelled L1 distances.

## Constraints

- $1 \leq T \leq 2$
- $1 \leq N \leq 35$
- $1 \leq M \leq 30$
- $1 \leq i \leq 6\,000$, for every square

## Scoring

- **input 1** : $T = 1$, $N \leq 5$, $M \leq 6$ and $i \leq 5$
- **input 2** : $T = 1$, $N \leq 10$, $M \leq 15$ and $i \leq 10$
- **input 3** : $T = 2$, $N \leq 20$, $M \leq 20$ and $i \leq 75$
- **input 4** : $T = 2$, $N \leq 30$, $M \leq 25$ and $i \leq 1\,200$
- **input 5** : $T = 2$, $N \leq 35$, $M \leq 30$ and $i \leq 6\,000$

## Examples

| input | output |
|---|---|
| 1<br>4 4<br>1 0 1<br>3 1 1<br>2 2 2<br>0 3 1 | Case #1: 10 |