

Supraiden

Duck is playing a game named "Supraiden", which is another version of a famous shooting game named "Raider". On a $D \times 10^9$ map, initially Duck is at the left top coordinates (1, 1), and there are N enemies at the bottom of the map but with different horizontal position L_i . That is, the first enemy is at (D, L_1) , the second one is at (D, L_2) , and so on. The i -th enemy has M_i bullets, and he will shoot the j -th bullet at time S_{ij} .

All enemies are fixed, so they cannot move; and Duck can choose to stay at the same position, or move one unit on map horizontally per unit of time. For example, moving from (1, 1) to (1, 5) takes 4 units of time. But Duck can only move to a cell where currently no bullets are there.

Assume Duck will move to right at next unit of time, the following are two examples will cause valid move, followed by two invalid move examples:

(D = Duck, E = enemy, ^ = bullet from enemy)

```
Valid          Invalid (Duck will get hit)
..D..  ..D^..  ..D..  ..D^..
.....  .^^..   ...^..  ...^..
...^..  .....  ...^..  ...^..
.EEE..  .EEE..  .EEE..  .EEE..
```

Bullet speed is one unit on map per unit of time. Shooting takes 0 unit of time, but Duck and enemies can only shoot one bullet per unit of time and shoot vertically. For example, moving from (1, 1) to (1, 5) and shoot immediately only takes 4 units of time. When two bullets collide with each other, both disappear. Collision only occurs when one side shoots before the bullet from another side reaches his current position.

Assume Duck will shoot at next unit of time, the following are two examples will cause valid collision, followed by two invalid collision examples:

(D = Duck, E = enemy, v = bullet from Duck, ^ = bullet from enemy)

```
Valid          Invalid (Duck will get hit)
..D..  ..D..  ..D..  ..D..
..v..  .....  ..^..  ..^..
...^..  ...^..  .....  ...^..
..E..  ..E..  ..E..  ..E..
```

Duck has infinite amount of bullets, starting at time 0, your task is find the minimum time to kill all enemies without getting hit.

Input

The first line is the number of test cases T . ($1 \leq T \leq 20$)

For each test case, it starts with two integers D, N . ($3 \leq D \leq 10^9, 1 \leq N \leq 8$)

Following N lines, each starts with L_i, M_i , followed by M_i distinct integers S_{ij} . ($1 \leq L_i \leq 10^9, 1 \leq M_i, \leq 1000, 1 \leq S_{ij} \leq 10^9$)

* L_i and S_{ij} are already sorted in ascending order.

Output

Output the minimum time to kill all enemies.

Example

Input:

```
3
100 1
1 1 100
5 3
12 9 0 1 4 5 6 7 8 9 19
14 2 12 88
20 5 23 27 29 35 100
6 2
6 7 1 2 3 4 5 6 7
8 1 0
```

Output:

```
99
29
18
```

Explanation

In case 1, Duck can shoot one bullet at time 0.

In case 2, Duck reaches (1, 12) at time 14, and shoots. Then he moves and reaches (1, 14) at time 17 and shoots. Lastly he reaches (1, 20) at 23 and shoots three bullets. The first two bullets collided with the first two bullets shot by enemy 3, and the last bullet hits the enemy at time 29.

In case 3, Duck reaches (1, 6) but doesn't shoot, then moves to (1, 8) immediately and shoots one bullet. After that, he goes back and reaches (1, 6) at time 13 and shoots. The bullet hits enemy 1 at time 18.