Move the bishop

Consider you have a chess board with NxN squares, 1 ≤ N ≤ 100.000.000. There is only a piece on the board: the bishop. The position of the bishop is given by a pair of numbers 1 ≤ r,c ≤ N; r is the row and c is the column. Position (1,1) refers to the bottom-left square of the board, while position (N,N) refers to the up-right square.

The problem consists of calculating the minimum number of movements that the bishop have to do to reach a given square of the board, given the position of the bishop and the position of that square. If this movement is not possible, the output must be: "no move". Don´t worry if you don´t know how to play chess. The only information you need is: the bishop moves diagonally any number of squares, forwards or backwards as long as its path is not blocked by other pieces.

The Input

The input begins with a single integer, C, indicating the number of test cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive test cases.

For each test case, the first line contains an integer 1 ≤ T ≤ 100, indicating the number of tests for that case. The second line contains an integer 1 ≤ N ≤ 100.000.000, for a chess board with NxN squares. Then the test lines follow, and for each one, there are four numbers separated by a single space. The first two numbers are the row and column where the bishop is, and the last two numbers are the row and column of the position of the square that the bishop have to reach.

The Output

For each test line you should produce an output line. This line just contains a number that indicates the minimum number of movements for the bishop according to positions described in the input line, or the message "no move" if the position is not reachable.

Sample Input

```
2

3
8
3 6 6 3
```

```
4 2 2 3
7 2 1 4

2
6
1 2 6 5
2 3 5 1
```

Sample Output

```
1
no move
2
2
no move
```