3985 Board Games

You have been hired by the quality control division of a world famous board game company. Their creative and design division comes up, on a daily basis, with great ideas for board games, but sometimes the scoring of the proposed games does not match the storyboard or leads the player to impossible or undesirable situations.

Most of the games this company produces can be described loosely as race games. Race games are games where the players need to go from an initial square to a final square, performing along the way, a series of moves, gaining or losing score points for each of those moves. Moves can be influenced by player's decisions, drawing of cards, rolling of dices, etc..

Your task is to check the description of a given game, stating the lowest possible score, or if it can lead to an infinitely high (there's no way the player can win the game), or to an infinitely low score.

Input

Input consists of multiple test cases. The first line of the input contains the number of test cases.

There is a blank line before each dataset.

Each dataset consists of one game description. The first line of the input contains a positive integer N not greater than 300, indicating the number of squares in the game. The second line contains two non-negative integers, I and F, defining the initial and final squares for this game, where I and F are lesser than N. The third line contains an integer M, indicating the number of possible moves of the game. The following M lines describe all the possible moves of the game. Each line, describing one possible move, consists of three integers, respectively, the initial square and final square of the move, in the range $[0, N-1]$, and the corresponding score.

Output

The output for each dataset consists of a single line with an integer, indicating the lowest possible score for the proposed game. If the scores are infinitely high or low then your program should output '`infinity`'.

Print a blank line between datasets.

Sample Input

```
2

4
0 3
4
0 1 5
0 2 7
2 1 -3
1 3 2

4
0 3
3
0 1 5
```

```
0 2 7
2 1 -3
```

Sample Output

```
6

infinity
```