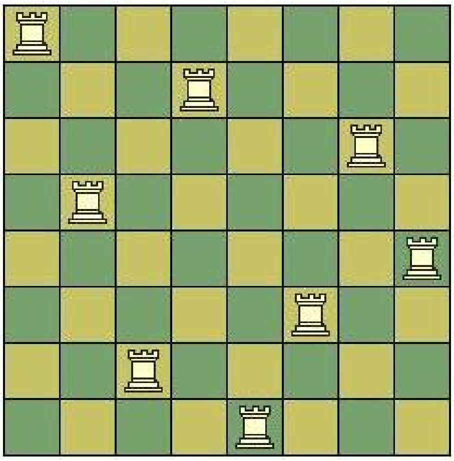
P15



We would like to place n rooks, $1 \leq n \leq 5000$, on a $n \times n$ board subject to the following restrictions

- The i-th rook can only be placed within the rectangle given by its left-upper corner (xl_i, yl_i) and its right-lower corner (xr_i, yr_i), where $1 \leq i \leq n$, $1 \leq xl_i \leq xr_i \leq n$, $1 \leq yl_i \leq yr_i \leq n$.

- No two rooks can attack each other, that is no two rooks can occupy the same column or the same row.

Input

The input consists of several test cases. The first line of each of them contains one integer number, n, the side of the board. n lines follow giving the rectangles where the rooks can be placed as described above. The i-th line among them gives xl_i, yl_i, xr_i, and yr_i. The input file is terminated with the integer '`0`' on a line by itself.

Output

Your task is to find such a placing of rooks that the above conditions are satisfied and then output n lines each giving the position of a rook in order in which their rectangles appeared in the input. If there are multiple solutions, any one will do. Output '`IMPOSSIBLE`' if there is no such placing of the rooks.

Sample Input

```
8
1 1 2 2
5 7 8 8
2 2 5 5
2 2 5 5
6 3 8 6
6 3 8 5
6 3 8 8
3 6 7 8
8
1 1 2 2
5 7 8 8
2 2 5 5
2 2 5 5
6 3 8 6
6 3 8 5
6 3 8 8
3 6 7 8
0
```

Sample Output

```
1 1
5 8
2 4
4 2
7 3
8 5
6 6
3 7
1 1
5 8
2 4
4 2
7 3
8 5
6 6
3 7
```

P16

You and your friend are playing a game in which you and your friend take turns removing stones from piles. Initially there are N piles with $a_1, a_2, a_3, \ldots, a_N$ number of stones. On each turn, a player must remove at least one stone from one pile but no more than half of the number of stones in that pile. The player who cannot make any moves is considered lost. For example, if there are three piles with 5, 1 and 2 stones, then the player can take 1 or 2 stones from first pile, no stone from second pile, and only 1 stone from third pile. Note that the player cannot take any stones from the second pile as 1 is more than half of 1 (the size of that pile). Assume that you and your friend play optimally and you play first, determine whether you have a winning move. You are said to have a winning move if after making that move, you can eventually win no matter what your friend does.

Input

The first line of input contains an integer T ($T \leq 100$) denoting the number of testcases. Each testcase begins with an integer N ($1 \leq N \leq 100$) the number of piles. The next line contains N integers $a_1, a_2, a_3, \ldots, a_N$ ($1 \leq a_i \leq 2 * 10^{18}$) the number of stones in each pile.

Output

For each testcase, print 'YES' (without quote) if you have a winning move, or 'NO' (without quote) if you don't have a winning move.

Sample Input

```
4
2
4 4
3
1 2 3
3
2 4 6
3
1 2 1
```

Sample Output

```
NO
YES
NO
YES
```