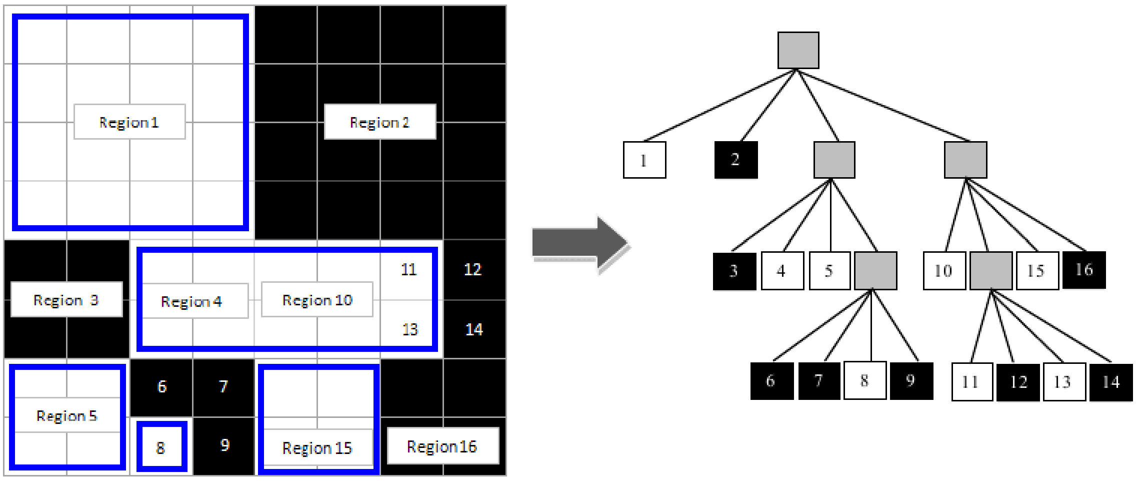
11941 Creating a Quadtree

A *quadtree*, first introduced by Finkel and Bentley, is a tree data-structure in which each internal node has exactly four children. Quadtrees are often used for problems that can be mapped into a two-dimensional space which is then recursively subdivided it into four equally-sized regions while a certain condition holds. Such problems include the following: collision detections, spatial data indexing, image compression and Conway's Game of Life. The problem consists in compressing a binary image represented by an $n \times n$ matrix, where $n > 1$ is a power of 2. The binary image is specified by means of rectangular areas containing only white pixels. Each area is given by the location of its upper-left and lower-right corners. The compressed image is given by a quadtree, where each node denotes a region in the image. A region is subdivided if it is not a pixel yet and it has both white and black pixels. The order in visiting regions is left-right, top-down.

For a better understanding of this problem, consider the third test case from Sample Input where the binary image is a 8×8 matrix composed by five white areas. In the following figure, such areas are highlighted with blue rectangles. Its corresponding quadtree is made up of five internal nodes and sixteen leaves (regions).



Input

The first line contains an integer $N > 0$ denoting the number of test cases. The next N lines contain a space-separated list starting with an integer $n > 1$, power of two, which denotes the matrix size, and followed by $m \geq 2$ pairs of integers (i_k, j_k), such that:

a) i_k is the column index

b) j_k is the row index

c) $1 \leq i_k, j_k \leq n$

d) Two consecutive pairs (i_{k-1}, j_{k-1}), (i_k, j_k), where k is even, denotes respectively the position of the upper-left corner and lower-right corner of the $1/2k$-th area

e) If m is odd, the pair (i_m, j_m) is ignored

Output

The output consists of N lines containing either the quadtree produced at each test case, or the text '`Size is invalid`' if the matrix size is not a power of 2. The quadtree is specified by means of a sequence of `0`'s and `1`'s. For each black node add a '`0`' to the output; for each white node add a '`1`'; for each gray node add a '`*`' and repeat this process for each child node. Do not consider the root in the output. Traverse the quadtree in pre-order.

Sample Input

```
3
4 (1,1) (1,1) (4,1) (4,1) (1,3) (2,4)
15
8 (1,1) (4,4) (3,5) (7,6) (1,7) (2,8) (3,8) (3,8) (5,7) (6,8)
```

Sample Output

```
*1000*010010
Size is invalid
10*011*0010*1*101010
```