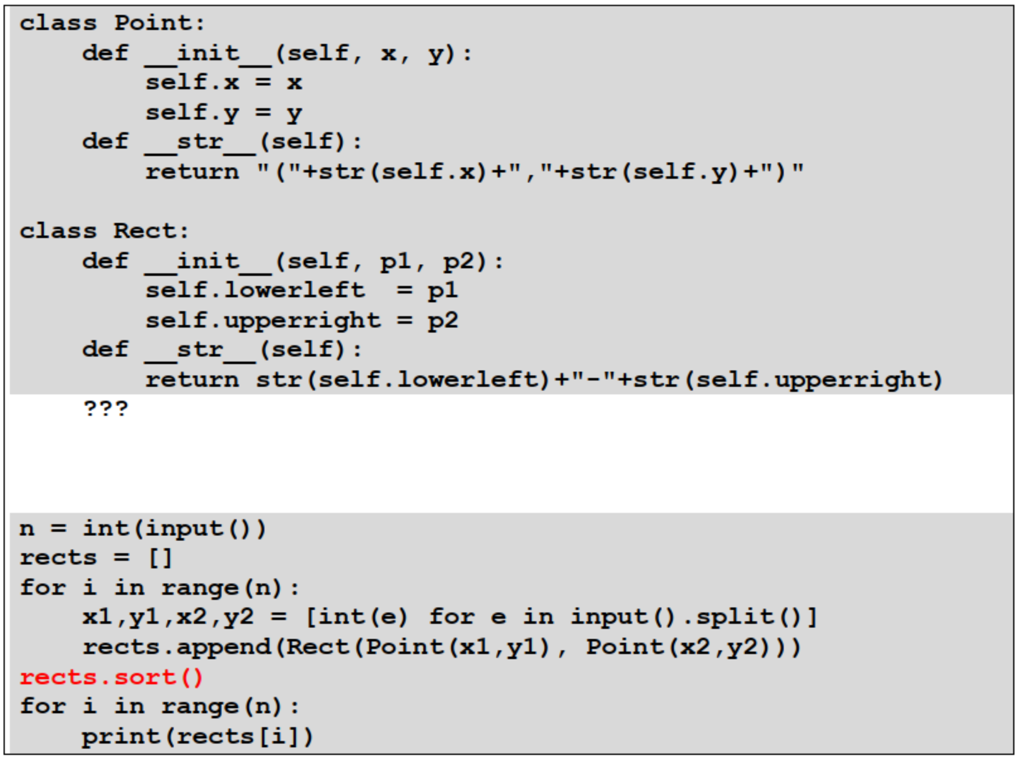
Sort rectangles by area

The picture below shows class **Point** which represents a point on 2-dimension plain. Class **Rect** represents a rectangle which contains a bottom-left point and top-right point of the rectangle (which sides are parallel with X-axis and Y-axis)

The program below receives rectangles to sort by area of them and display as output.



```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __str__(self):
        return "("+str(self.x)+","+str(self.y)+")"

class Rect:
    def __init__(self, p1, p2):
        self.lowerleft  = p1
        self.upperright = p2
    def __str__(self):
        return str(self.lowerleft)+"-"+str(self.upperright)
    ???



n = int(input())
rects = []
for i in range(n):
    x1,y1,x2,y2 = [int(e) for e in input().split()]
    rects.append(Rect(Point(x1,y1), Point(x2,y2)))
rects.sort()
for i in range(n):
    print(rects[i])
```

Your task is to make class **Rect** can be compared to each other by the area of them. If they can be compared, their list will be sorted easily by the program above. (You don’t have to change anything on the grey area.)

Input

The first line is an integer m which is a number of rectangles.

For the next m lines, each line contains 4 integer which represent (x,y) position of a bottom-left point and top-right point of the rectangle.

Output

Order of the rectangles sorted by area ascendingly.

Example

Input (from keyboard)	Output (on screen)

3
1 1 3 3
0 0 10 10
2 2 3 3
	(2,2)-(3,3)
	(1,1)-(3,3)
	(0,0)-(10,10)