

เรียงลำดับสี่เหลี่ยมผืนผ้าตามพื้นที่

ข้างล่างนี้แสดงคลาส **Point** แทนจุดในระนาบสองมิติ และมีคลาส **Rect** แทนสี่เหลี่ยมผืนผ้าที่ภายในเก็บจุดที่มุมซ้ายล่างกับมุมขวาบนของสี่เหลี่ยมผืนผ้า (มีด้านที่ขนานกับแกน **x** หรือแกน **y**) โปรแกรมข้างล่างนี้อ่านสี่เหลี่ยมเข้ามาเพื่อต้องการนำมาเรียงลำดับตามพื้นที่ของสี่เหลี่ยมจากน้อยไปมาก แล้วก็แสดงผลทางจอภาพ

```
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])
```

สิ่งที่ต้องการให้เขียนคือ ปรับปรุงคลาส **Rect** ที่ทำให้น่าอึดใจเจตต์ของสี่เหลี่ยมนี้มาเปรียบเทียบกันได้ว่าใครน้อยใครมาก โดยใช้พื้นที่เป็นตัวเปรียบเทียบ เมื่อทำได้เช่นนี้ จะทำให้เราสามารถ sort list of Rectangle objects ได้ง่าย ๆ ตามโปรแกรมที่เขียนข้างบนนี้ (ให้เติมเฉพาะบริเวณสีเทา อย่าแก้ไขบริเวณสีเทา)

ข้อมูลนำเข้า

บรรทัดแรกมีจำนวนเต็ม **n** แทนจำนวนสี่เหลี่ยมที่จะตามมา

n บรรทัดต่อมา แต่ละบรรทัดมีจำนวนเต็ม 4 จำนวนแทนพิกัด **x,y** ของมุมซ้ายล่างกับขวาบนของสี่เหลี่ยม

ข้อมูลส่งออก

ลำดับของสี่เหลี่ยมเรียงตามพื้นที่จากน้อยไปมาก

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
3	(2,2) - (3,3)
1 1 3 3	(1,1) - (3,3)
0 0 10 10	(0,0) - (10,10)
2 2 3 3	