



Master algorithms together on [Binary Search](#)! Create a room, invite your friends, and race to finish the problems.

Daily Coding Problem

[Blog](#)

Daily Coding Problem #185

Problem

This problem was asked by Google.

Given two rectangles on a 2D graph, return the area of their intersection. If the rectangles don't intersect, return 0.

For example, given the following rectangles:

```
{
  "top_left": (1, 4),
  "dimensions": (3, 3) # width, height
}
```

and

```
{
  "top_left": (0, 5),
  "dimensions": (4, 3) # width, height
}
```

return 6.

Solution

To determine the intersection of the two rectangles, we find the right-mostmost left border, bottom-most top border, left-most right border, and top-most bottom border.

If any of these pairs don't cross each other then we return 0, since they don't intersect. Otherwise, we now have the coordinates of the intersecting rectangle, so we can compute its area.

```
def rectangles(rec1, rec2):
    left_x = max(rec1["top_left"][0], rec2["top_left"][0])
    right_x = min(rec1["top_left"][0] + rec1["dimensions"][0], rec2["top_left"]
[0] + rec2["dimensions"][0])

    top_y = min(rec1["top_left"][1], rec2["top_left"][1])
    bottom_y = max(rec1["top_left"][1] - rec1["dimensions"][1], rec2["top_left"]
[1] - rec2["dimensions"][1])

    if left_x > right_x or bottom_y > top_y:
        return 0

    return (right_x - left_x) * (top_y - bottom_y)
```

This runs in $O(1)$ time and space.

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)