1 line Ruby, 2 lines Python

StefanPochmann ★ 47270 June 11, 2016 8:52 PM 5.3K VIEWS

16

Idea: Reflect the points by replacing every x with minX+maxX-x and then check whether you get the same points. Why minX+maxX-x? I actually thought of it as minX+(maxX-x), i.e., first the subtraction (maxX-x). That's how far x is away from the max, so instead go that distance from the min.

Update to reflect the changed problem: (Originally, the problem was about a set of points, so no duplicates.)

```
def is_reflected(points)
  points.sort!.uniq == points.map { |x, y| [points[0][0] + points[-1][0] - x, y] }.sort.uniq
end
```

```
def isReflected(self, points):
    points = sorted(set(map(tuple, points)))
    return points == sorted((points[0][0] + points[-1][0] - x, y)
                            for x, y in points)
```

Ruby

```
def is_reflected(points)
  points.sort! == points.map { |x, y| [points[0][0] + points[-1][0] - x, y] }.sort
end
```

Python

```
def isReflected(self, points):
    points.sort()
    return points == sorted([points[0][0] + points[-1][0] - x, y]
                            for x, y in points)
```

A linear time one:

```
def isReflected(self, points):
    if not points: return True
    X = min(points)[0] + max(points)[0]
    return {(x, y) for x, y in points} == {(X - x, y) for x, y in points}
```

Shorter, but I think less nice:

```
    return set(map(tuple, points)) == {(X - x, y) for x, y in points}
```

python ruby short

Type comment here... (Markdown is supported)

Post

orbuluh ★ 1283 June 13, 2016 2:55 AM