Problem # 497 : Random Points in Non-overlapping Rectangles

------------------------------------------------------------------------------------

<https://leetcode.com/problems/random-point-in-non-overlapping-rectangles/discuss/806567/Simple-Python-3-solution-with-explanation>

1. The number of points in each rectangle = (x2 - x1 + 1) \* (y2 - y1 + 1)

2. I calculated the cumulative density of the number of points from the rectangles using a running total of the number of points from each rectangle

3. The total number of points, num\_points = the last cumulative density value

4. Generate a random number between 1 and num\_points

5. Using get\_index, find out which rectangle this random number came from.

6. Get the specific rectangle, generate x\_rand between x1 and x2, and y\_rand between y1 and y2. The random point picked is [x\_rand, y\_rand]

class Solution:

def \_\_init\_\_(self, rects: List[List[int]]):

self.rects = rects

self.cum\_density = []

self.pts = 0 #cumulative number of points

for rect in rects:

x1, y1, x2, y2 = rect[0], rect[1], rect[2], rect[3]

self.pts += (x2 - x1 + 1) \* (y2 - y1 + 1)

self.cum\_density.append(self.pts)

def get\_index(self, random\_num):

n = len(self.rects)

for i in range(n):

if self.cum\_density[i] >= random\_num:

return(i)

def pick(self) -> List[int]:

num\_points = self.cum\_density[-1]

random\_num = random.randint(1, num\_points)

rect\_index = self.get\_index(random\_num)

rect = self.rects[rect\_index]

x1, y1, x2, y2 = rect[0], rect[1], rect[2], rect[3]

x\_rand = random.randint(x1, x2)

y\_rand = random.randint(y1, y2)

return([x\_rand, y\_rand])

# Your Solution object will be instantiated and called as such:

# obj = Solution(rects)

# param\_1 = obj.pick()