Problem # 1465: Maximum Area of a Piece of Cake After Horizontal and Vertical Cuts (Medium)

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

<https://leetcode.com/problems/maximum-area-of-a-piece-of-cake-after-horizontal-and-vertical-cuts/>

My Solution:

Runtime beats 79.74%

1. Add 0 and h to the horizontalCuts list and similarly 0 and w to the verticalCuts list.

2. Sort both lists.

3. Initialize max\_h and max\_w to 0.

4. Let m be the length of the horizontalCuts list and n the length of the verticalCuts list.

5. Iterate through the horizontalCuts list from the element at index 1 and find the difference in value of the elements at the index and the preceding index. Then update max\_h if the difference is larger than the existing max\_h.

6. Similarly iterate through the verticalCuts list from the element at index 1 and find the difference in value of the elements at the index and the preceding index. Then update max\_w if the difference is larger than the existing max\_w.

7. Return the product of max\_h and max\_w modulo 10^9 + 7

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

class Solution:

def maxArea(self, h: int, w: int, horizontalCuts: List[int], verticalCuts: List[int]) -> int:

# Append 0 and h to horizontalCuts, and 0 and w to verticalCuts

horizontalCuts = [0] + horizontalCuts + [h]

verticalCuts = [0] + verticalCuts + [w]

# Sort the lists

horizontalCuts = sorted(horizontalCuts)

verticalCuts = sorted(verticalCuts)

# Initialize max\_h and max\_w with 0

max\_h = 0 #initialize

max\_w = 0 # initialize

m = len(horizontalCuts)

n = len(verticalCuts)

# iterate through horizontalCuts to find the max height

# between 2 xconsecutive lines

for i in range(1, m):

piece\_h = horizontalCuts[i] - horizontalCuts[i - 1]

max\_h = max(max\_h, piece\_h)

# iterate through verticalCuts to find the max width

# between 2 xconsecutive lines

for j in range(1, n):

piece\_w = verticalCuts[j] - verticalCuts[j - 1]

max\_w = max(max\_w, piece\_w)

# Return the area formed by max height and max width

# Remember to use modulo to the area.

return (max\_h \* max\_w) % (10\*\*9 + 7)