Problem # 977: Square of a Sorted Array (Easy)

https://leetcode.com/problems/squares-of-a-sorted-array/

My Solution:

1. Let the length of the array nums be n.

2. Initialize the left pointer to point to element at index 0 and the right pointer to point to element at index n -1.

3. Create an array res of length n populated with None.

4. Set index for res as n – 1.

5. While the left pointer is less than or equal to the right pointer, compare the absolute values of nums at index left and at index right. Take the larger one and assign its squared value to res array at index i. If nums at index left is larger, then increment left index by 1. If nums at index right is larger, then decrement the right index by 1. At the end of the iteration, decrement i by 1.

6. Return the res array.

NOTE: The approach is to populate the result array res from right to left and from largest to the smallest value.

class Solution:

def sortedSquares(self, nums: List[int]) -> List[int]:

n = len(nums)

left = 0

right = n -1

res = [None] \* n

i = n - 1

while left <= right:

if abs(nums[left]) >= abs(nums[right]):

res[i] = nums[left] \* nums[left]

left += 1

else:

res[i] = nums[right] \* nums[right]

right -= 1

i -= 1

return res