Problem #5: Longest Palindrome Substring. (Medium)

https://leetcode.com/problems/longest-palindromic-substring/

My solution:

Runtime beats 12.48%

1. Let n be the length of s. If n is 1, then return s.

2. Otherwise set res to be the first character in the string (of length 1).

3. Iterate with i from index 0 through n, and j from n through i decrementing by 1.

Consider the substring from index from index i through index j.

4. If the length of the substring is less than res, then break.

There is no point in checking if smaller substrings are palindromes.

5. If the substring is greater than res in length, then check whether it is a palindrome.

If the substring is a palindrom, then break from the inner loop since there is no point in considering a smaller substring.

6. Finally return res.

class Solution:

def longestPalindrome(self, s: str) -> str:

n = len(s)

if n == 1:

return s

res = s[0]

for i in range(n):

for j in range(n, i, -1):

substring = s[i:j]

if len(substring) > len(res):

if substring == substring[::-1]:

res = substring

break

else:

break

return res