class Solution:

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

        if s[::-1] == s:

            return s

        # inti start at 0 and window size is 1

        start, size = 0, 1

        for i in range(1, len(s)):

            # set up l, r pointer , base on cur idx to left

            # cur window size = max window size + 1

            l, r = i - size, i + 1

            # cur window size = max window size + 2

            s1 = s[l-1:r]

            # if s1 is palindrome, update start idx and the size

            if l >= 1 and s1[::-1] == s1:

                size += 2

                start = l - 1

            else:

                # cur window size = max window size + 1

                s2 = s[l:r]

                if s2[::-1] == s2:

                    size += 1

                    start = l

        return s[start:start+size]