5. longest palindromic substring

Approach1: time: o(n2) space: o(n2) 🡪 o(n) dynamic programming

A piece of paper with writing on it

Description automatically generated with medium confidence

Approach2: time: o(n2) space: o(1)

A palindromic substring is a substring that read same from both directions.

We will get the longest length, initially set to zero.

Then we will set the initial left and right pointer of our result to 0.

Now we are going to go through every single character in this string, and considering it to be the center.

So im going to have a left and right pointer. These left and right pointers are going to be initialized to i which is our center positions right now.

While the left and right pointers are in bound as well as while this is a palindrome. So I want to check that the left and right are equal to each other. We are starting in the middle and expanding outwards. So while this is the case, we know this is palindrome. Then we check the length of this palindrome which can computed by r-l +1

If the current length of the palindrome is greater than longest length, we are going to update the longest length, and left and right pointer of results. After that we need expand our pointer outward, left -1 and right +1.

There is also a corner case for even length of palindrome. There is no center for the palindrome.

So im going to set left pointer to i and right pointer to i+1. Then we basically copy and paste the code we write above. We can also write a function to do it.

Finally, we return the longest palindrome substring.