Longest Palindrome in a string

BRUTE FORCE :

Consider all substrings and check whether it is palindrome or not.

- Time Complexity : O(n^3)
- Space Complexity : O(1)

Better Approach :

Instead of considering all the string from start we can try to choose a char and consider it as the middle element and move left as well as right simultaneously to check if the letter on the current letter's left and right are same if not we break from the loop and consider the next char. We need to keep in mind that a palindromic string can either be of even or odd length so we run a loop considering left and right both to be at i considering odd length and moving left and right simultaneously. We also run a loop for odd length considering left to be at i and right to be at i+1 and then moving on both sides. We maintain a global variable reslen and res which stores the maximum length of palindromic string and string respectively.

Code :

```
string longestPalinSubstring(string str)
{
    // Write your code here.
    int n=str.size(),reslen=0,left,right;
    string res;
    for(int i=0;i<n;i++)
    {
        left=i,right=i;
        while(left>=0 && right<n && str[left]==str[right])
        {
            if(reslen<right-left+1)
            {
                res=str.substr(left,right-left+1);
                reslen=right-left+1;

            }
            left--;
            right++;
        }
        left=i,right=i+1;
        while(left>=0 && right<n && str[left]==str[right])
        {
            if(reslen<right-left+1)
            {
                res=str.substr(left,right-left+1);
                reslen=right-left+1;

            }
            left--;
            right++;
        }

    }
    return res;
}
```

- Time Complexity : O(n^2)
- Space Complexity : O(1)

Another Approach : DP