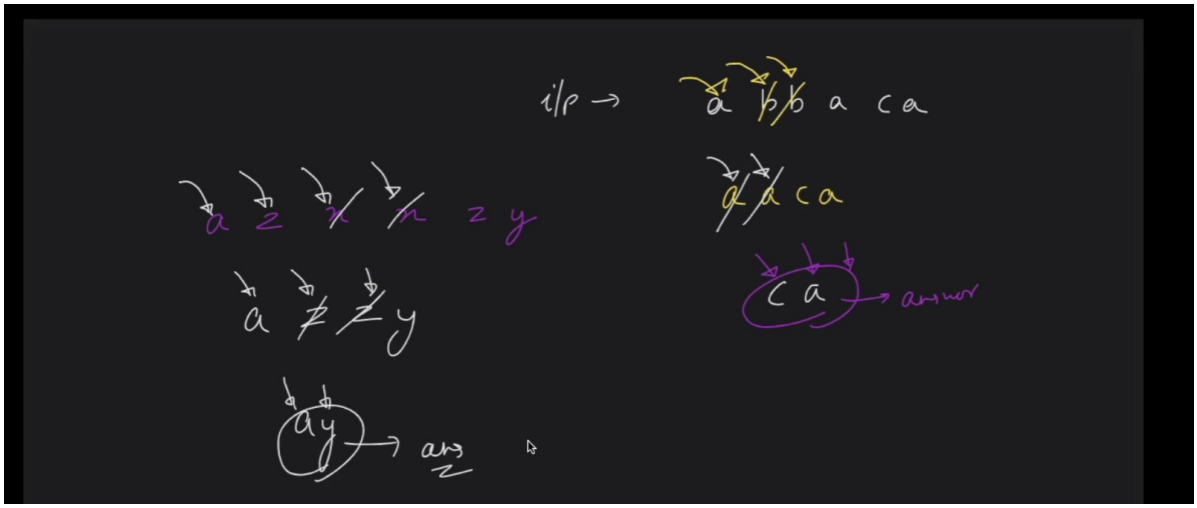
Ch-5(L-2) : String

Classwork Questions

▼ **Remove all adjacent duplicates in string**

Question :



Logic :

1. phele empty string banao called `ans`
2. agr ans ka last char , string ke current char ke equal h to pop out krdo i.e `ans[ans.length()-1] == s[i]` → `pop_back()`
3. nhi to push_back krna

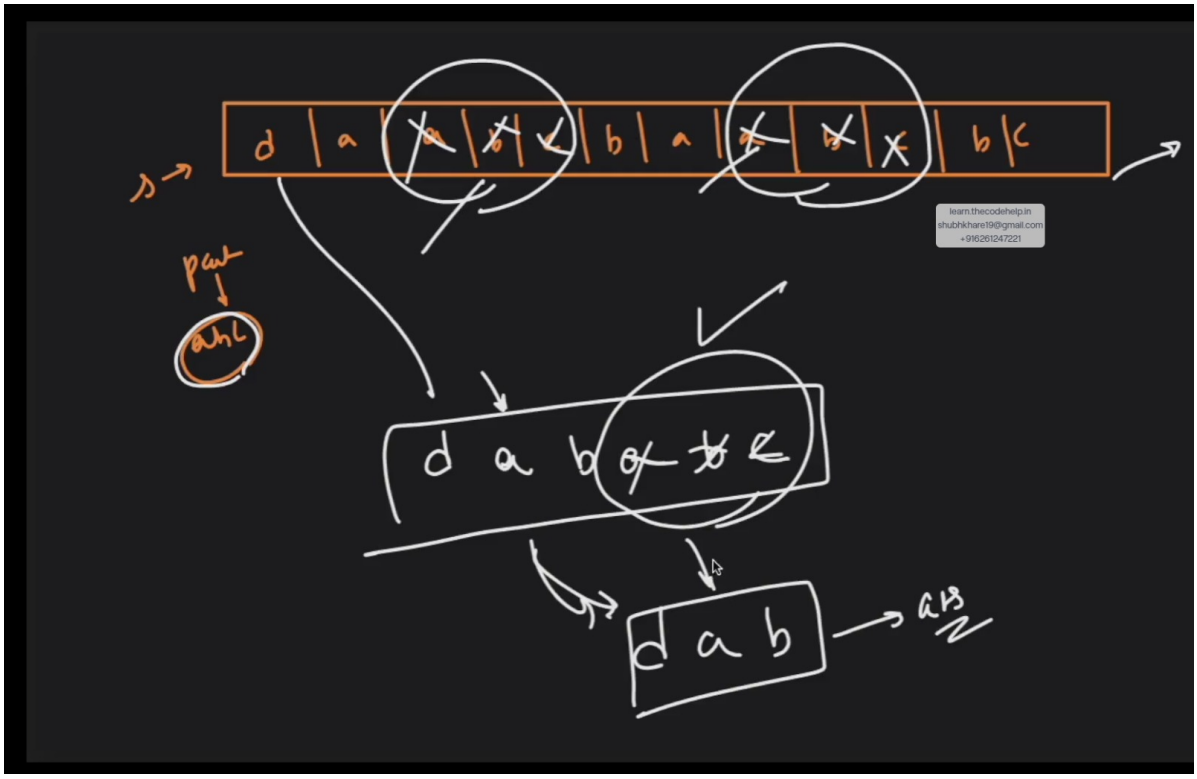
Code :

```
class Solution {
public:
    string removeDuplicates(string s) {
       string ans = "";
```

```
        int i =0;
        while(i < s.length()){
             // string is not empty
            if(ans.length() > 0){
                if(ans[ans.length()-1] == s[i]){
                    ans.pop_back();
                }
                else{
                    ans.push_back(s[i]);
                }
            }
             // string is empty
            else{
                ans.push_back(s[i]);
            }
            i++;
        }
        return ans;
    }
};
```

▼ **Remove all the Occurrences of a Sub - String**

Question :



Logic :

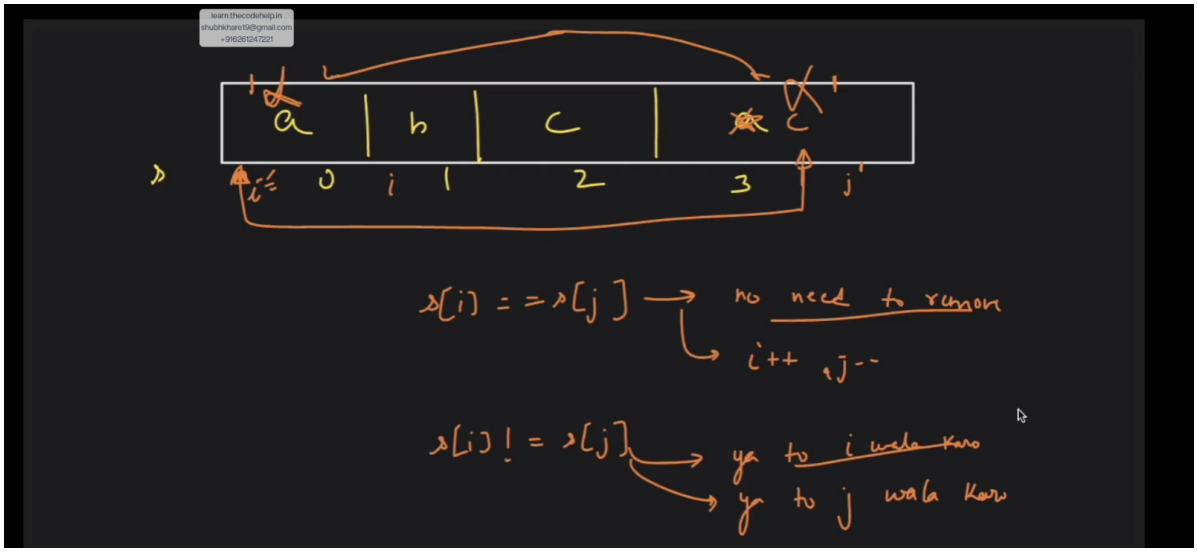
1. `Find` can be used to find the sub-string
2. `erase` it can be used to remove element

Code :

```
class Solution {
public:
    string removeOccurrences(string s, string part) {
        int pos = s.find(part);
        while(pos != string::npos){
            s.erase(pos,part.length());
            pos = s.find(part);
        }
        return s;
    }
};
```

▼ **Valid Palindrome II**

Question :



Logic :

1. `s[i] == s[j]` → no need to remove, i++, j - -
2. `s[i] != s[j]`

a. i remove krdo then i+1 → j tak palindrome check krruga

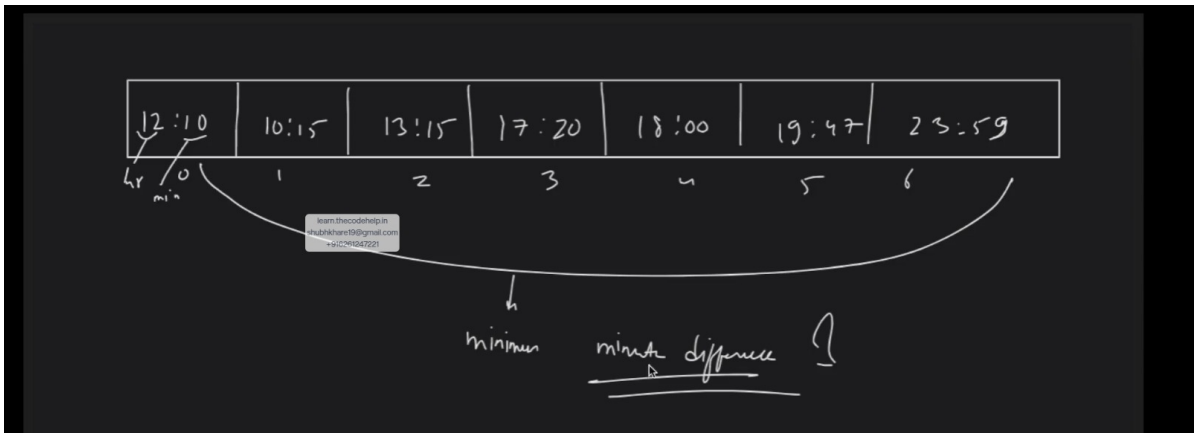
b. j remove krdo then i → j-1 tak palindrome check krruga

Code :

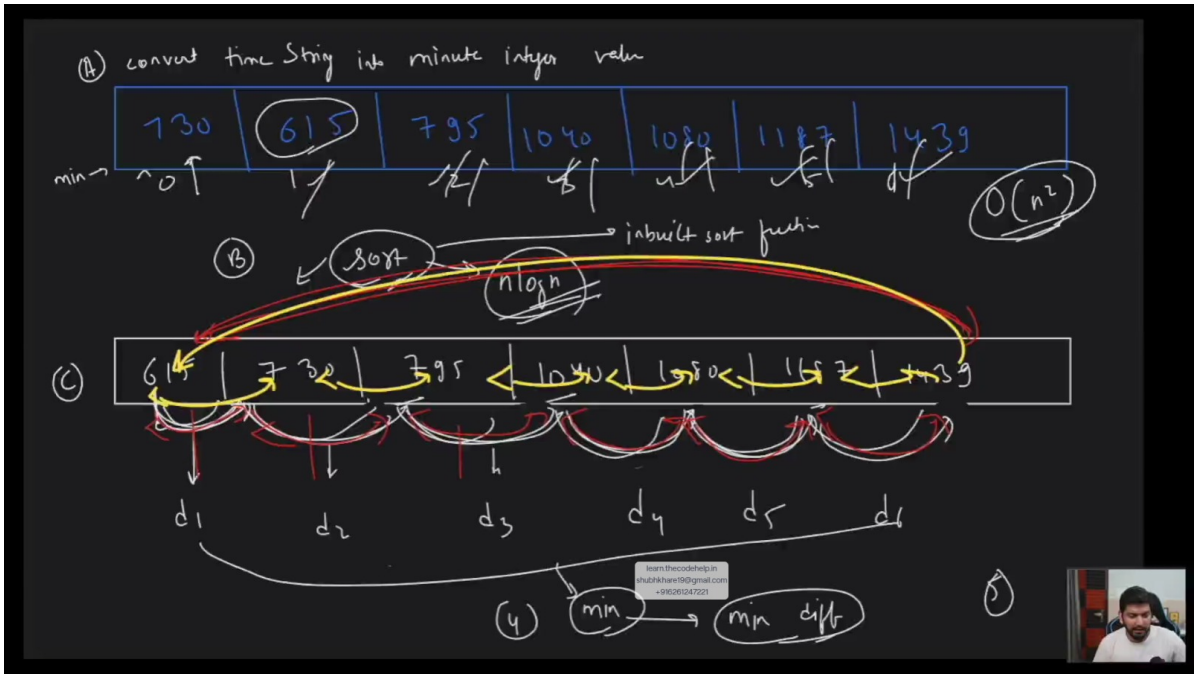
```
class Solution {
public:
    bool checkPalindrome(string s,int i, int j){
        while (i <= j){
            if(s[i] != s[j]){
                return false;
            }
            i++;
            j--;
        }
        return true;
    }
    bool validPalindrome(string s) {
        int i = 0;
        int j = s.length()-1;
        while(i <= j){
            if(s[i] != s[j]){
                // ya to i ko remove krdo ya fir j ko remove krdo
                return checkPalindrome(s, i+1, j) || checkPalindrome(s, i,j-1);
            }
            else{
                // s[i] == s[j]
                i++;
                j--;
            }
        }
        return true;
    }
};
```

▼ **Minimum Time Difference**

Question :



Logic :



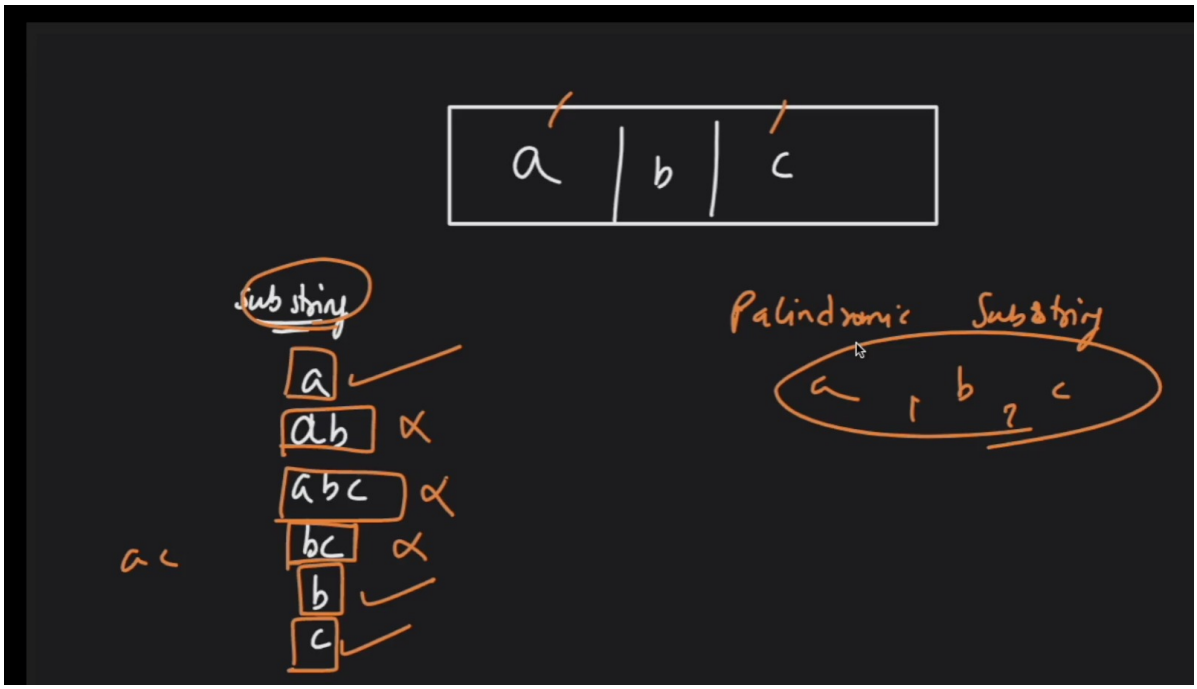
1. Convert all the time (HH:MM) `[string]` into minutes `[int]` using `stoi()` → string to int function
2. sort the minutes array then find all the difference between the adjacent elements
3. find the minimum difference among

Code :

```
class Solution {
public:
    int findMinDifference(vector<string>& timePoints) {
        // step 1 : converting into minutes
        vector <int> minutes;

        for(int i=0; i<timePoints.size(); i++){
            string curr = timePoints[i];

            int hr = stoi( curr.substr(0,2) );
            int min = stoi ( curr.substr(3,2) );
            int totalMin = hr * 60 + min;
            minutes.push_back(totalMin);
        }

        // step 2 : sort
        sort(minutes.begin(),minutes.end());

        // step 3 : difference and calculate minimum difference
        int mini = INT_MAX;
        int n =minutes.size();
        for(int i=0; i<n-1; i++){
            int diff = minutes[i+1] - minutes[i];
            mini = min(diff,mini);
        }

        // time circular hota h, last difference bi dekhna padega - THIS IS THE GAME
        int lastDiff1 = (minutes[0] + 1440) - minutes[n-1];
        int lastDiff2 =  minutes[n-1] - minutes[0];
        int lastDiff = min(lastDiff1,lastDiff2);
        mini = min(mini,lastDiff);
        return mini;
    }
};
```

▼ Palindromic Substrings

Question :



logic :

Method -1 :

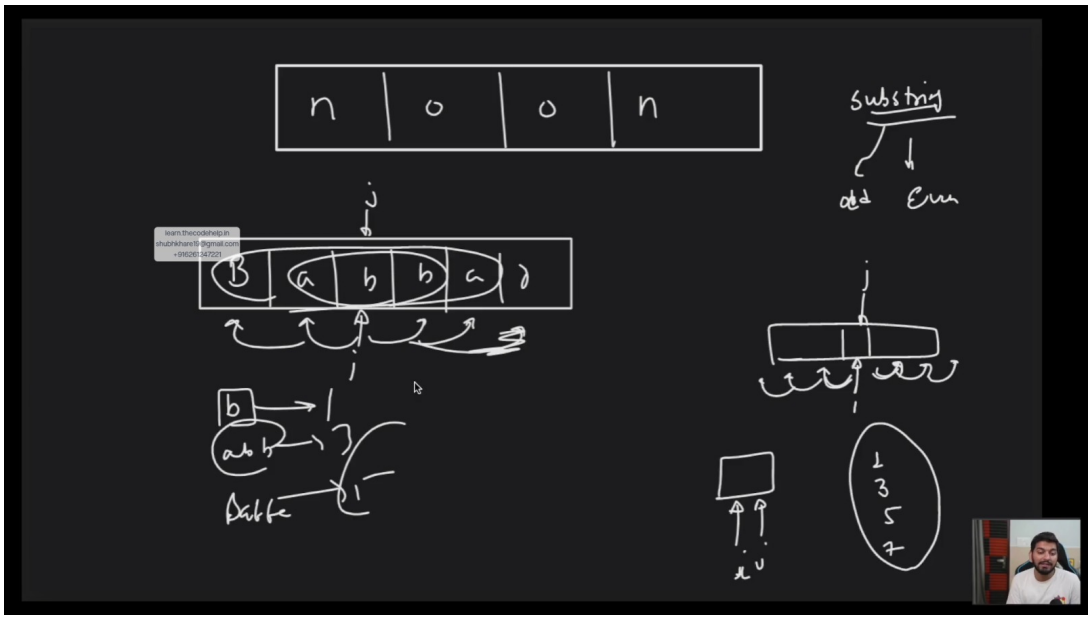
TC is O(n^3)

1. find all the substring
2. check for palindrome and count

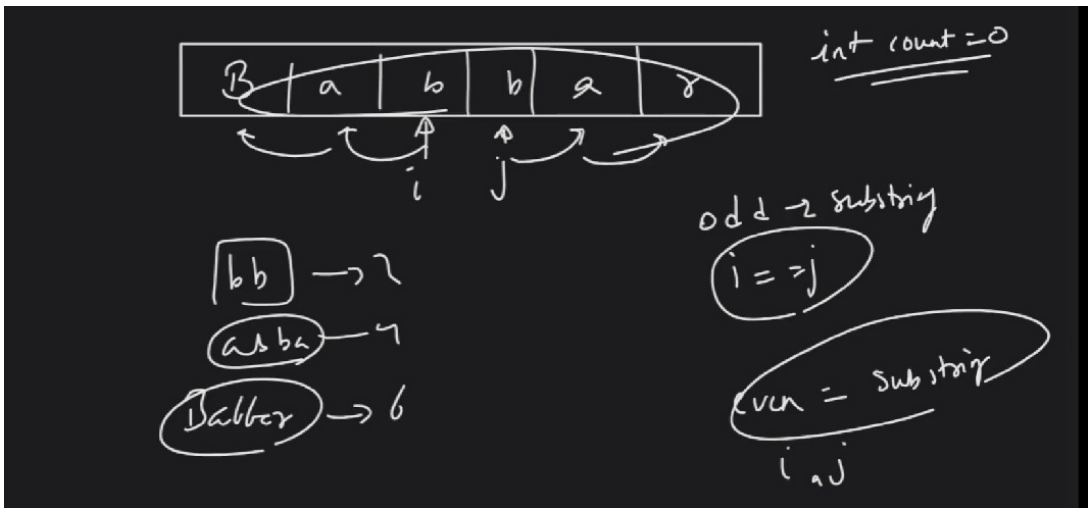
Method - 2 :

TC is O(n^2)

▼ When i and j ko same position par rakhte ho to aap odd length ki sub-string nikal pate ho



▼ When i and j ko adjacent position par rakhte ho to aap even length ki sub-string nikal pate ho



Code :

```
class Solution {
public:
    int expandAroundIndex(string s,int left, int right){
        int count =0;
        while(left>=0 && right<s.length() && s[left]==s[right]){
            count ++;
            left--;
            right++;
```

```
        }
        return count;
    }
    int countSubstrings(string s) {
        int totalCount = 0;
        for(int centre=0; centre<s.length(); centre++){
            // odd
            int oddAns = expandAroundIndex(s,centre,centre);
            totalCount = totalCount + oddAns;
            // even
            int EvenAns = expandAroundIndex(s,centre,centre+1);
            totalCount = totalCount + EvenAns;
        }
        return totalCount;
    }
};
```

Homework Question :

- ▼ Write the logic of erase and find function