Question: https://leetcode.com/problems/count-different-palindromic-subsequences/

Explanation: https://www.youtube.com/watch?v=fvYlinirmFg

Code:  
class Solution {

public int countPalindromicSubsequences(String s) {

HashMap<Character, Integer> mem = new HashMap<>();

int prev[]=new int[s.length()], next[]=new int[s.length()];

long dp[][]=new long[s.length()][s.length()];

//keeping track of prev occurence of charecters

for(int i=0; i<s.length(); i++){

char c=s.charAt(i);

if(!mem.containsKey(c)){

prev[i]=-1;

}else{

prev[i]=mem.get(c);

}

mem.put(c, i);

}

mem.clear();

//keeping track of next occurence of charecters

for(int i=s.length()-1; i>=0; i--){

char c=s.charAt(i);

if(!mem.containsKey(c)){

next[i]=s.length();

}else{

next[i]=mem.get(c);

}

mem.put(c, i);

}

//now check using gap approach

for(int gap=0; gap<s.length(); gap++){

for(int i=0, j=gap; j<s.length(); j++, i++){

if(gap==0){

dp[i][j]=1;

}else if(gap==1){

dp[i][j]=2;

}else{

char sc = s.charAt(i), ec = s.charAt(j);

if(sc!=ec){//if starting and ending characters are different

dp[i][j]=dp[i+1][j]+dp[i][j-1]-dp[i+1][j-1];

}else{//if starting and ending characters are same

int np=next[i], pp=prev[j];

if(np>pp){//if they dont have a similar character in between i and j pos

dp[i][j]=2\*(dp[i+1][j-1])+2;

}else if(np==pp){//if they have a similar character in between i and j pos once

dp[i][j]=2\*(dp[i+1][j-1])+1;

}else if(np<pp){//if they have similar characters in between i and j pos

dp[i][j]=2\*(dp[i+1][j-1])-dp[np+1][pp-1];

}

}

}

if (dp[i][j] < 0) {

dp[i][j] += 1000000007;

} else {

dp[i][j] %= 1000000007;

}

}

}

return (int)(dp[0][s.length()-1]);

}

}

Github Link :<https://lnkd.in/ecwtJeaz>