Question: https://leetcode.com/problems/minimum-insertion-steps-to-make-a-string-palindrome/

Initially the problem seems to be quite difficult, but think carefully when can a string be palindrome?

It can be palindrome if s and reverse(s) has Longest common subsequence equal to n, i.e., s.length().

So find L.C.S of s and reverse(s),i.e., x.

Now n-x characters are not part of L.C.S.

So we need to add n-x elements to make it a palindrome.

Code:  
class Solution {

public int minInsertions(String s) {

int n = s.length();

int[][] dp = new int[n+1][n+1];

for (int i = 0; i < n; ++i)

for (int j = 0; j < n; ++j)

dp[i + 1][j + 1] = s.charAt(i) == s.charAt(n - 1 - j) ? dp[i][j] + 1 : Math.max(dp[i][j + 1], dp[i + 1][j]);

return n - dp[n][n];

}

}

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