DP Concepts & Questions

Video 26

"HARDWORK HAS NO SUBSTITUTE..."

भाषण
(Motivation)

[cswithMIK → Twitter
Facebook, Instagram → codestorywithMIK
whatsapp → codestorywithMIK]

Done
- 1-D based DP
- 2-D based DP

Progress
- String based DP
- Grid based DP
- Game Strategy

We'll do :-
- RECURSION + MEMOIZATION (Top Down)
- Bottom UP
- Time & Space

DP on Strings :-

→ Longest Common Subsequence (LCS)

→ Print LCS

→ Edit Distance

→ Shortest common Supersequence. (SCS)

→ Print SCS

⟹ Pallindrome related DP problems :-

- Palindromic Substrings + Blueprint
- Longest Palindromic Substring

(•) Recursion + Memoization

(•) Bottom UP → Blue Print

1312. Minimum Insertion Steps to Make a String Palindrome

Hard | Topics | Companies | Hint

Given a string `s`. In one step you can insert any character at any index of the string.

Return *the minimum number of steps* to make `s` palindrome.

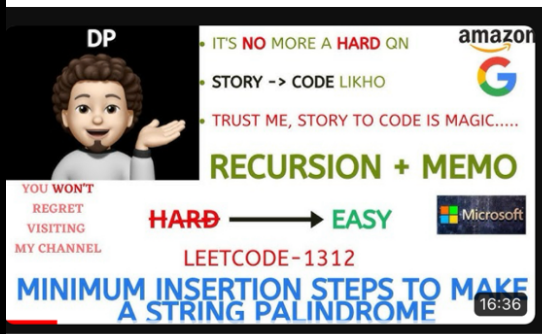
A **Palindrome String** is one that reads the same backward as well as forward.

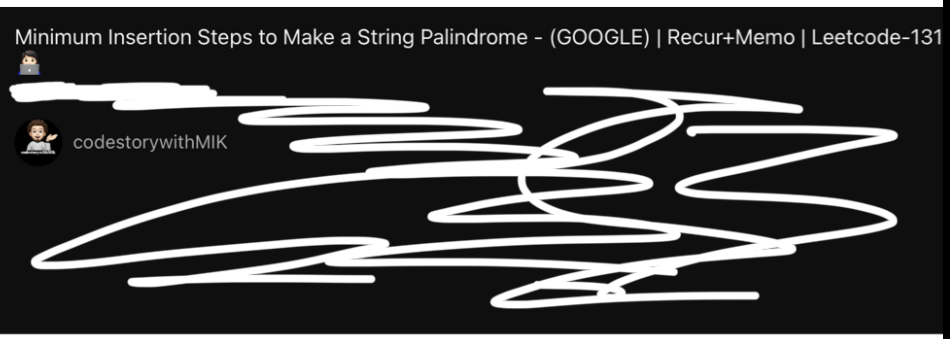
Example :- S = "mbadm"

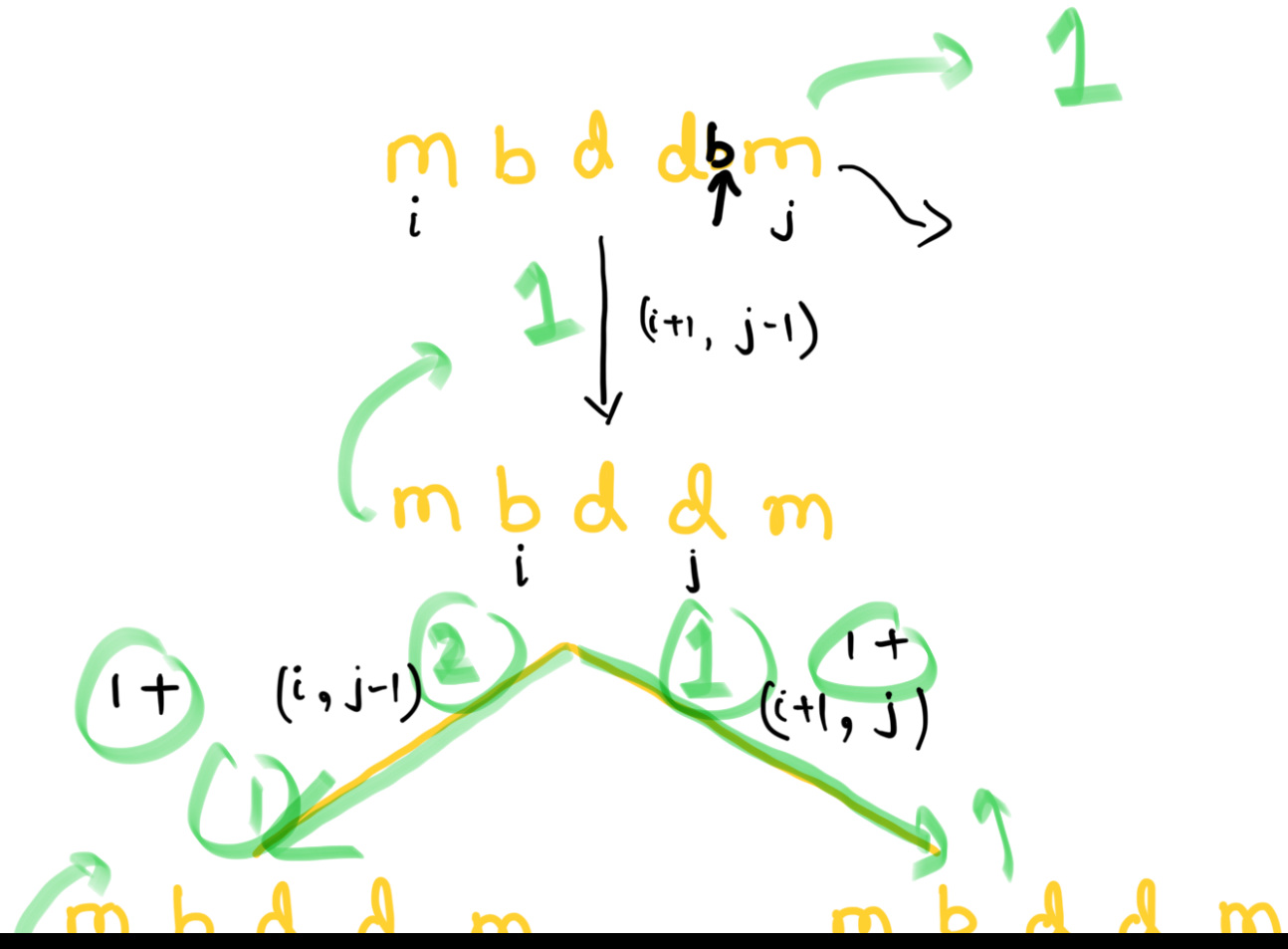
Output = 2

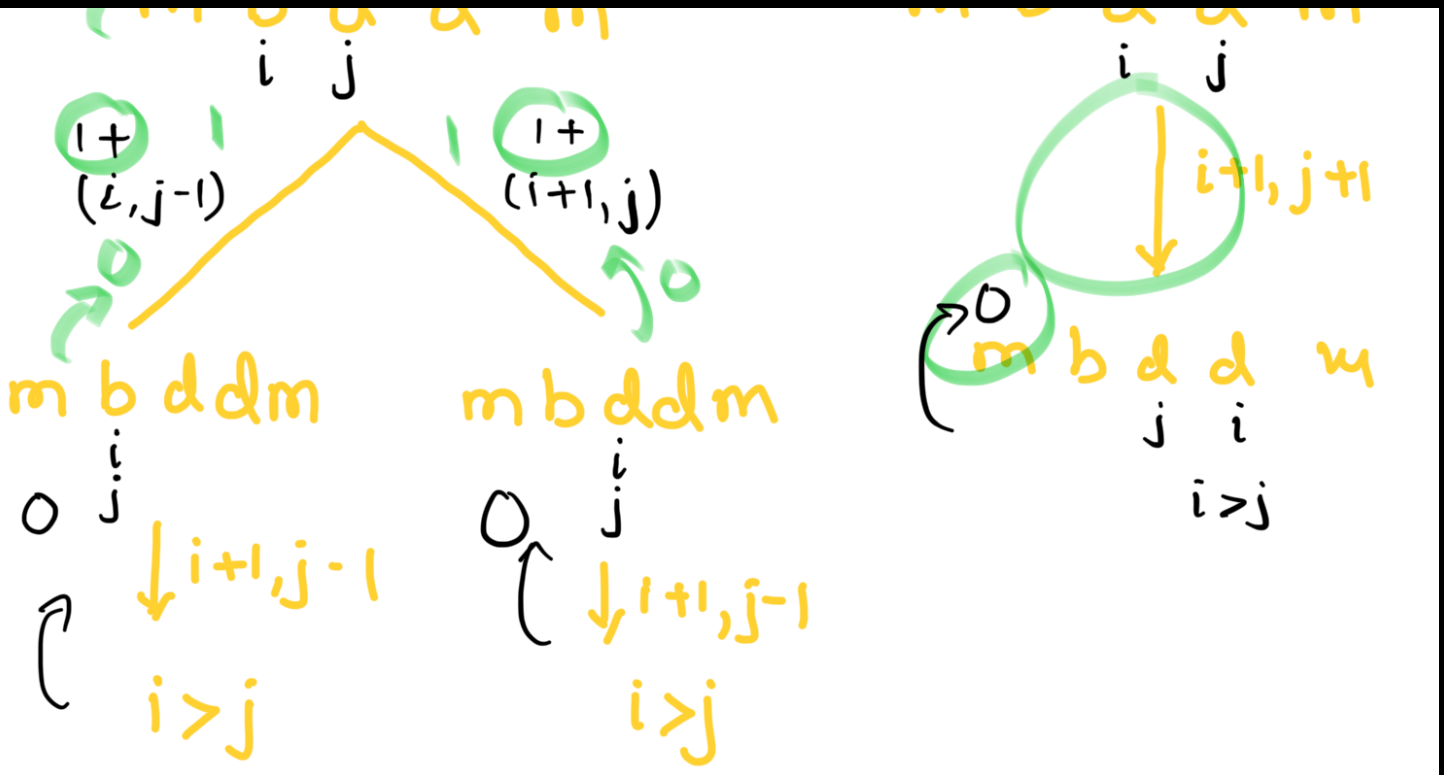
Why DP ??

Recursion + Moization









Using Bottom UP

(Blueprint)

```
class Solution {
public:
    int t[501][501];
    int solve(int i, int j, string &s) {

        if(i >= j)
            return 0;

        if(t[i][j] != -1)
            return t[i][j];

        if(s[i] == s[j])
            return t[i][j] = solve(i+1, j-1, s);

        return t[i][j] = 1 + min(solve(i, j-1, s), solve(i+1, j, s));

    }
```

S → 0 to n-1

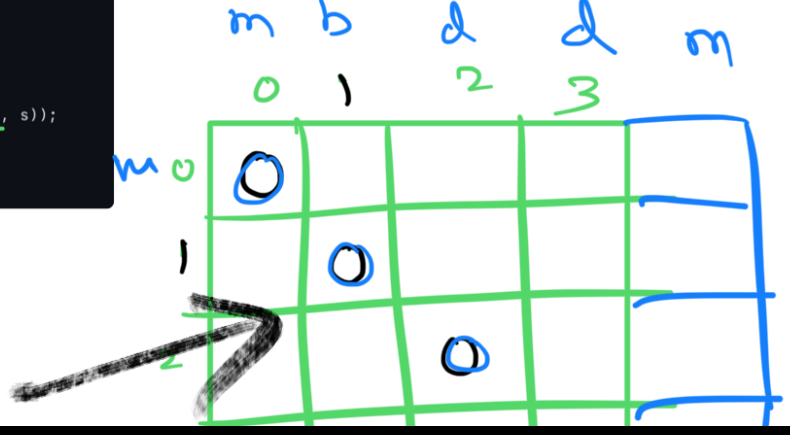
t[0][n-1]

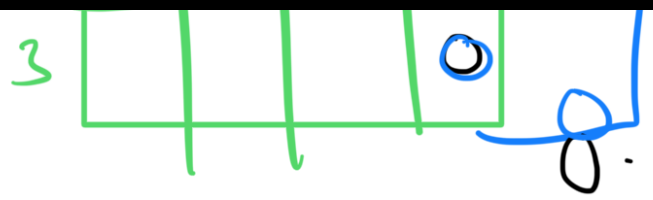
t[i][j] = min insertion to make s[i....j] a Palindrome.

m b d d m

0 1 2 3 4

[0...0]





Blue print code:-

t[ ][ ]

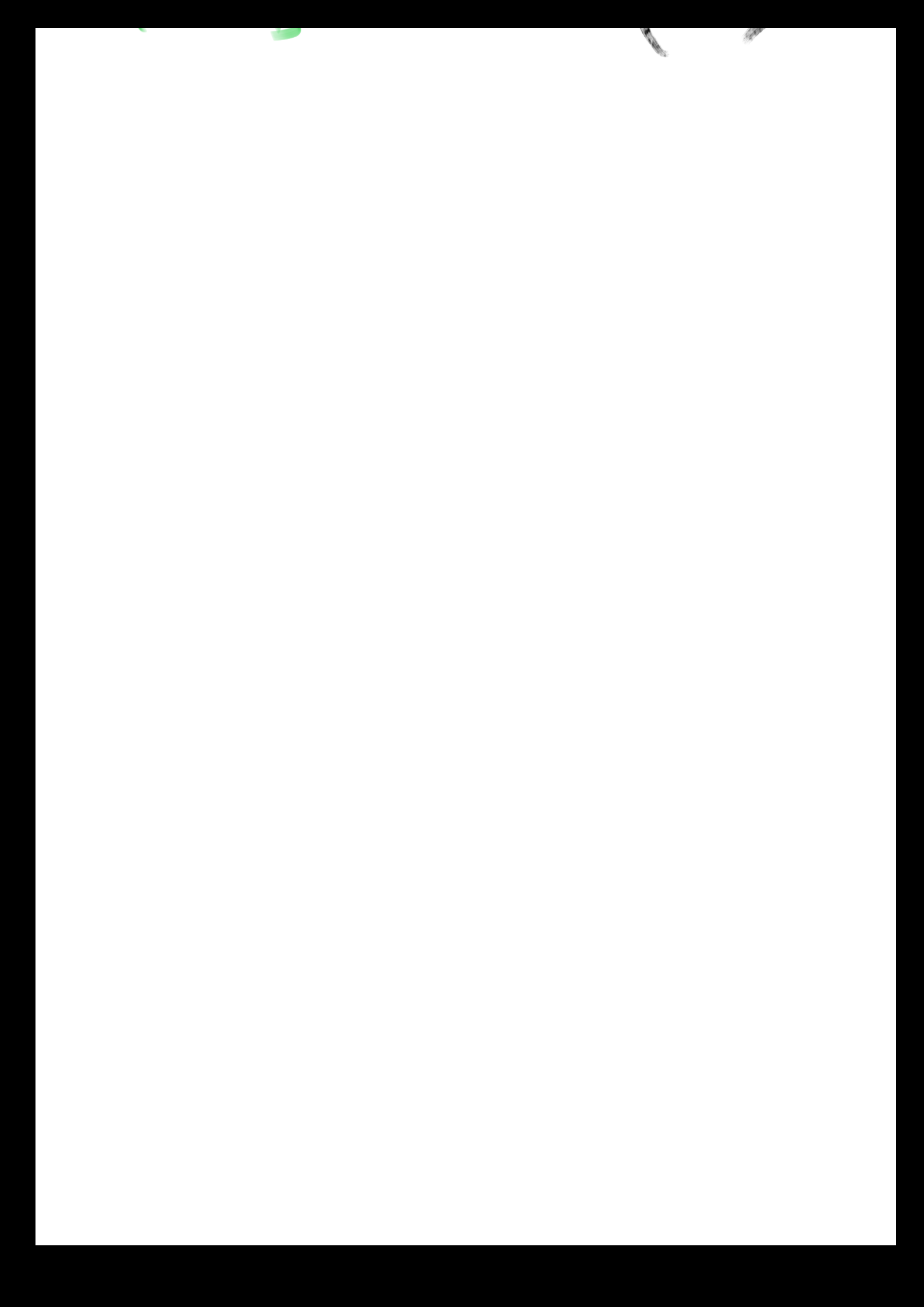
" a b c d "
i j

i+L = 0+2 = 2-1 = 1

```
for(int L=2; L<=n; L++) {
    for (i=0; i<n-L+1; i++) {
        j = i+L-1;
        if(s[i] == s[j]) { //no insertion
            t[i][j] = t[i+1][j-1];
        } else {
            t[i][j] = 1+min(t[i][j-1],
                            t[i+1][j]);
        }
    }
}
```

return t[0][n-1];

T.C = O(n²)

S[0...n-1]

S.C = O(n²)