DSA SERIES

- Learn Coding

Topic to be Covered today

Dynamic Programming

926. Flip String to Monotone Increasing

```
class Solution {
public:
    int n;


    int solve(string& s, int index, int prev , vector<vector<int>> &t) {
        if (index >= n)
            return 0;


        int flip = INT_MAX;
        int not_flip = INT_MAX;


        if(t[index][prev] !=-1){
            return t[index][prev];
        }


        if (s[index] == '0') {
            if (prev == 1) {
```

```
flip = 1 + solve(s, index + 1, 1,t);
        } else {
            flip = 1 + solve(s, index + 1, 1,t);
            not_flip = solve(s, index + 1, 0,t);
        }
    } else if (s[index] == '1') {
        if (prev == 1) {
            not_flip = solve(s, index + 1, 1,t);


        } else {


            flip = 1 + solve(s, index + 1, 0,t);
            not_flip = solve(s, index + 1, 1,t);
        }
    }


    return t[index][prev] = min(flip, not_flip);
}
```

```
    int minFlipsMonoIncr(string s) {
        n = s.length();


        vector<vector<int>> t(n+1 , vector<int>(2,-1));


        return solve(s, 0, 0,t);
    }
};
```

1312. Minimum Insertion Steps to Make a String Palindrome

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


        if (i >= j)
            return 0;


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


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


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

```
int minInsertions(string s) {
        int n = s.length();
        memset(t, -1, sizeof(t));


        return solve(s, 0, n - 1);
    }
};
```

Learn coding

THANK YOU