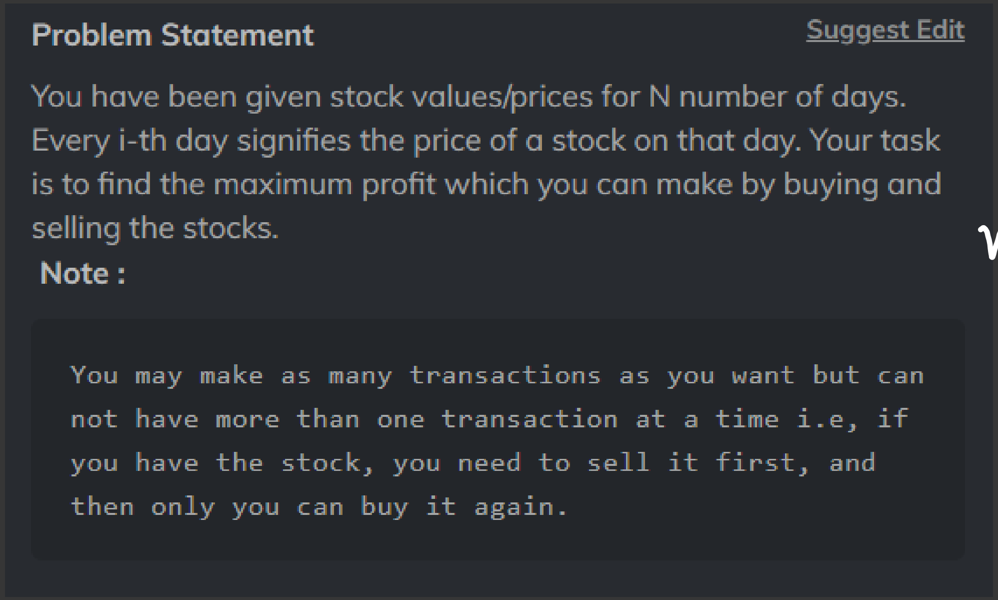
# DP on Stocks

## 1. Best Time to By & Sell Stock

arr[] = { 7, 1, 5, 3, 6, 4 }

if you are selling on $i^{th}$ day,
you buy on the minim price from 1st → (i-1)

# Buy and Sell Stock-II



**Problem Statement** Suggest Edit

You have been given stock values/prices for N number of days. Every i-th day signifies the price of a stock on that day. Your task is to find the maximum profit which you can make by buying and selling the stocks.

**Note :**

```
You may make as many transactions as you want but can
not have more than one transaction at a time i.e, if
you have the stock, you need to sell it first, and
then only you can buy it again.
```

Sample Input:

val = { 1 2 3 4 5 6 7 };

you can buy as many times I want

and sell as many times

## Understanding and Approach:

$$\text{prices} = \{7\ \underbrace{\overset{b}{1}\ \ \overset{s}{5}}_{4},\ \underbrace{\overset{b}{3}\ \ \overset{s}{6}}_{+3 = 7}\ \ 4\}$$

↑

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 7 | 1 | 5 | 3 | 6 | 4 |

↑

f(0,1)

↓

starting on 0th day with buy, what maximum profit?

**Recursion.**

f(i, buy)

# Recursion Solution:

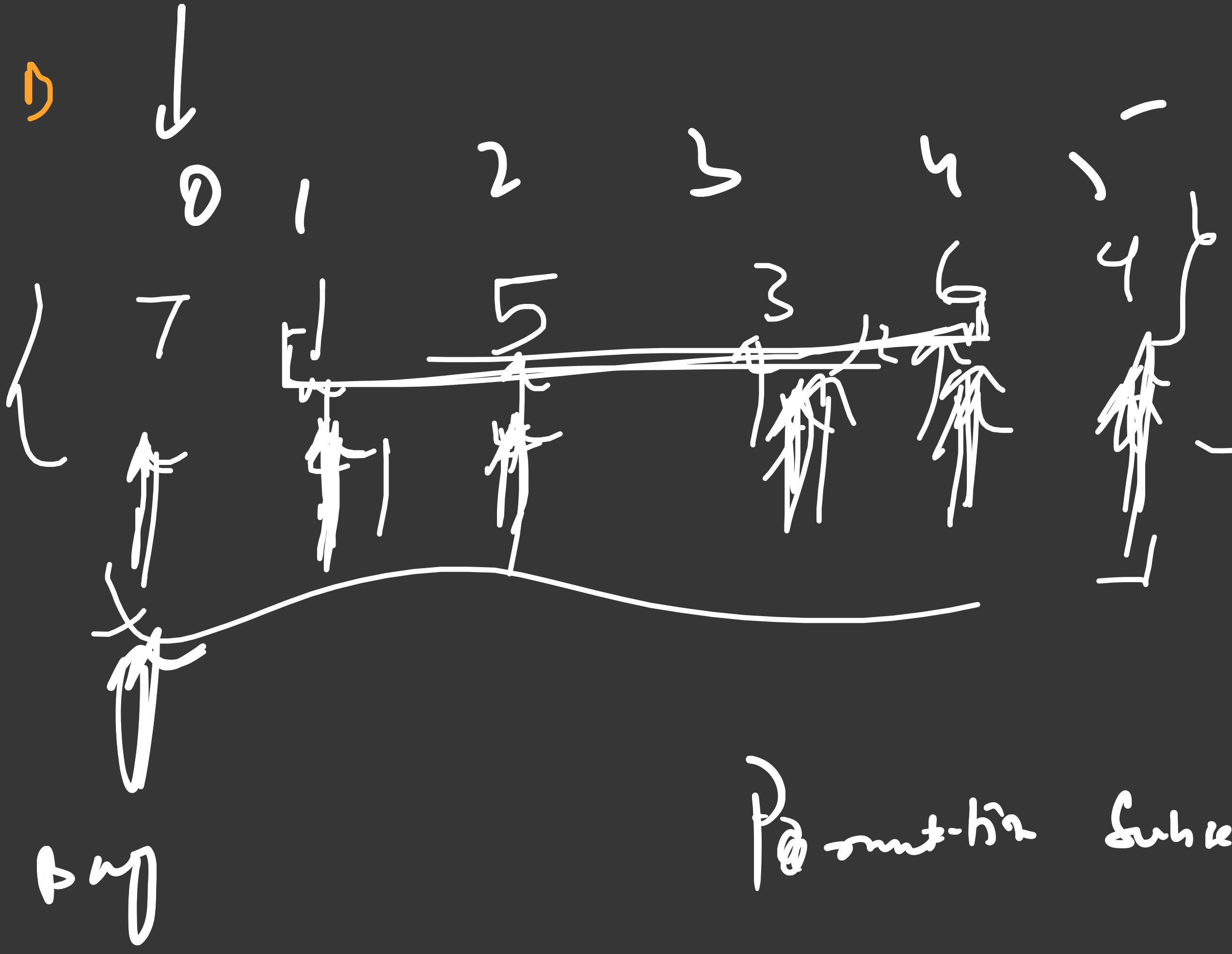
```
f(i, buy)
base case { if(i == n) return 0,

if (buy)
    profit = max { price [ind] + f(i+1, 0) } take
                   { 0 + f(i+1, 1) } not take

else
    profit = max { price [i] + f(i+1, 1),
                   0 + f(i+1, 0) }

return profit.
```

Approach:

at any and as many time

and sell it any day

as many time



→ Perm

Try all ways

Recursion

Permutation subseq – taken or not taken.

(I) Express all in terms of index

$f(i, buy)$

(II) Try all ways

(III) Max of all ways

(IV) Base Case.

sell

price[i] [i-

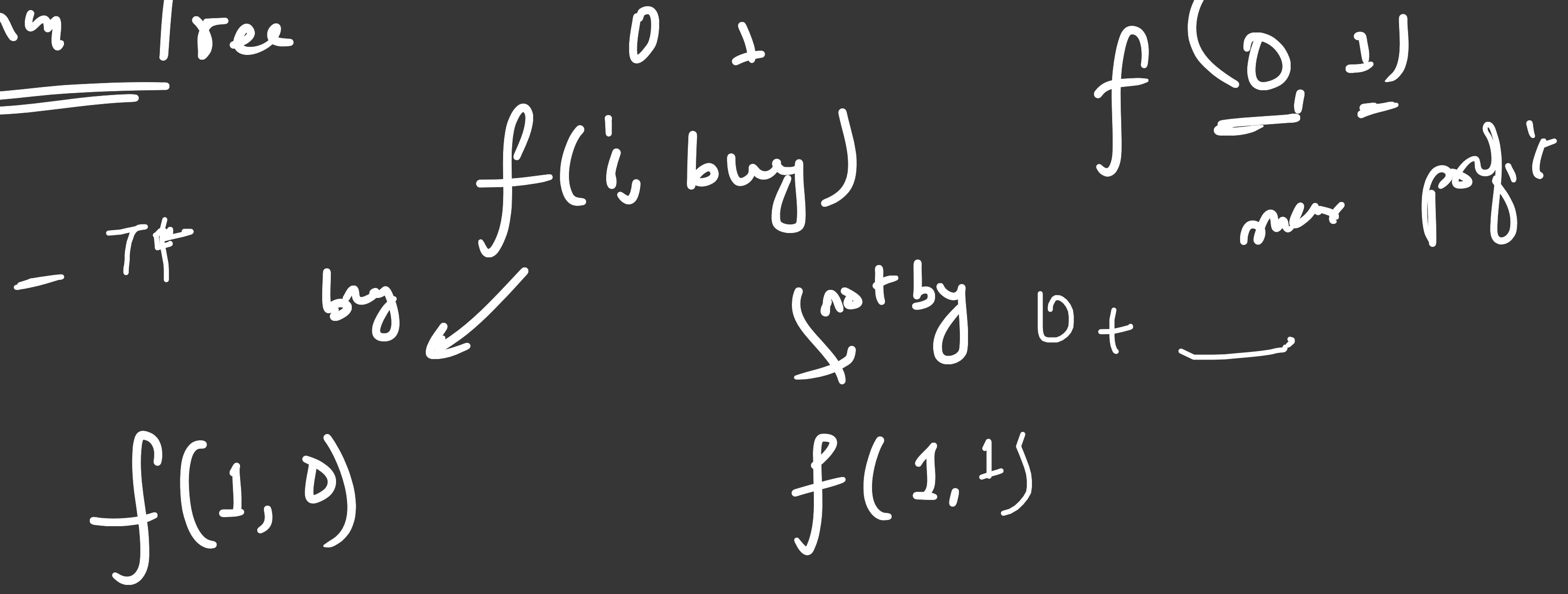
if Buy } sell[i] - buy[i]

profit ~ max {

else {

}

# Recursin Tree



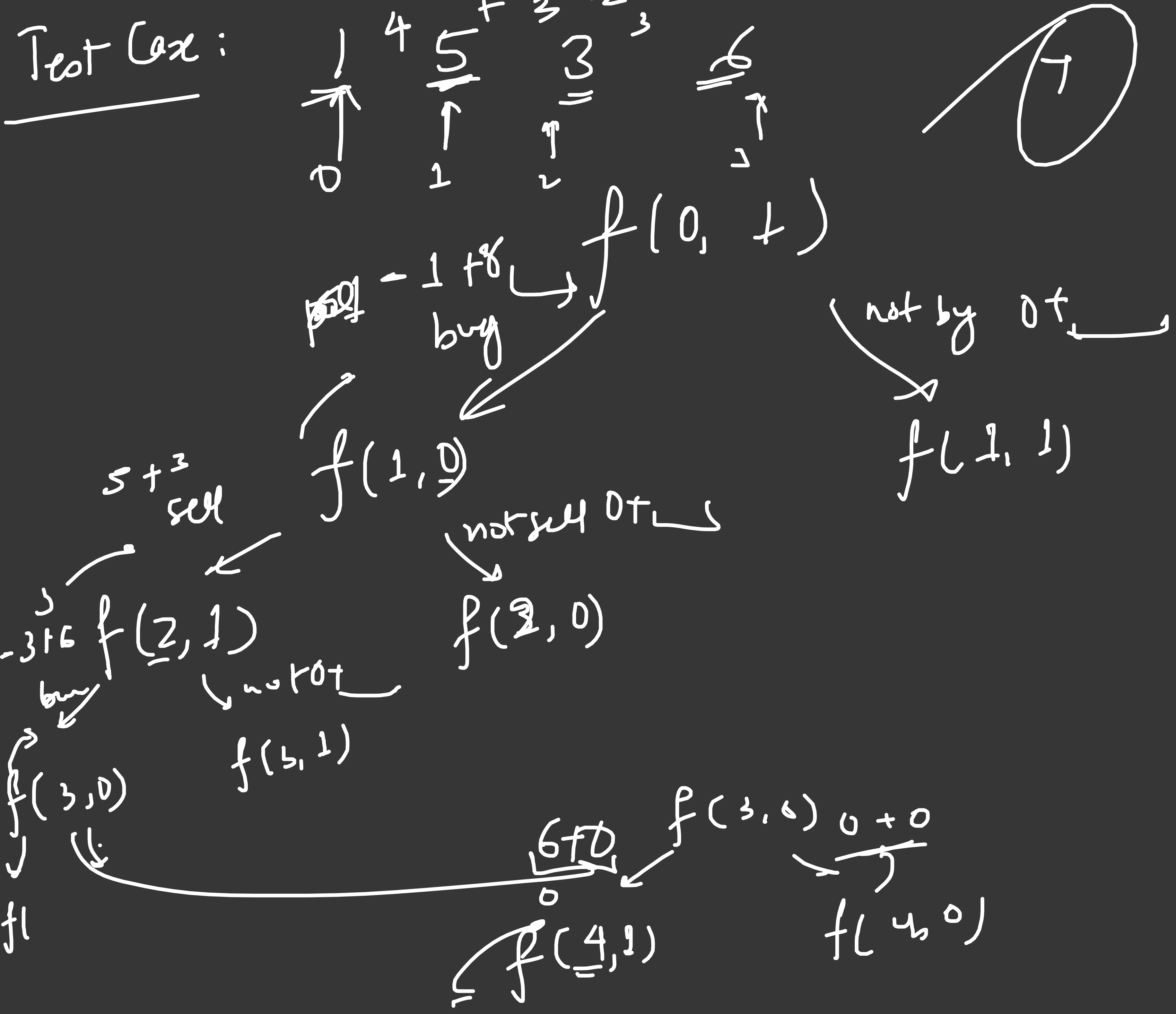
By true // ~~Buf~~ buy // not buy

profit = max $\left(\underline{-price[i] + f(i+1, 0)}, \; 0 + \underline{f(i+1, 1)}\right)$;

By Buy false // sell // not sell

profit = max ( price[i] + f( i+1, 1) , 0 + f(i+1, 0)

return profit


Test Case:
1 4 5 +3 3 2 3 6
0 1 2 3
f(0, 1)
-1 +8
buy
not by 0+
f(1,1)
f(1,0)
5+3
sell
not sell 0+
f(2,1)
-3+6
buy
not 0+
f(2,0)
f(3,0)
f(3,1)
f1
f(3,0)
6+0
0+0
0
f(4,1)
f(4,0)

## Memoization

f( i, buy)

dp [n+1]

dp [i] [buy]

**Base Case :**

dp [n] = 0,

## Tabulation:

Base Case ⟶

i

buy

Copy the Recurrence

```
for (i= n-1; i>=0; i--)
    for(int buy = 0, buy<1; buy++
```

## Space Optimisation:

36 → space Optimisation with variable and more

37

38

39

40

→

T:30

8 → 10 → Code Chef Contest

11:00 → 12:00

DP.

12:00 → 2:00 → Android

1 → 3:00 → DSA sheet