CLASS 85

15 May 2023 21:12

Class 85

Dynamic Programming

Buy and sell stocks with k transaction

Hard

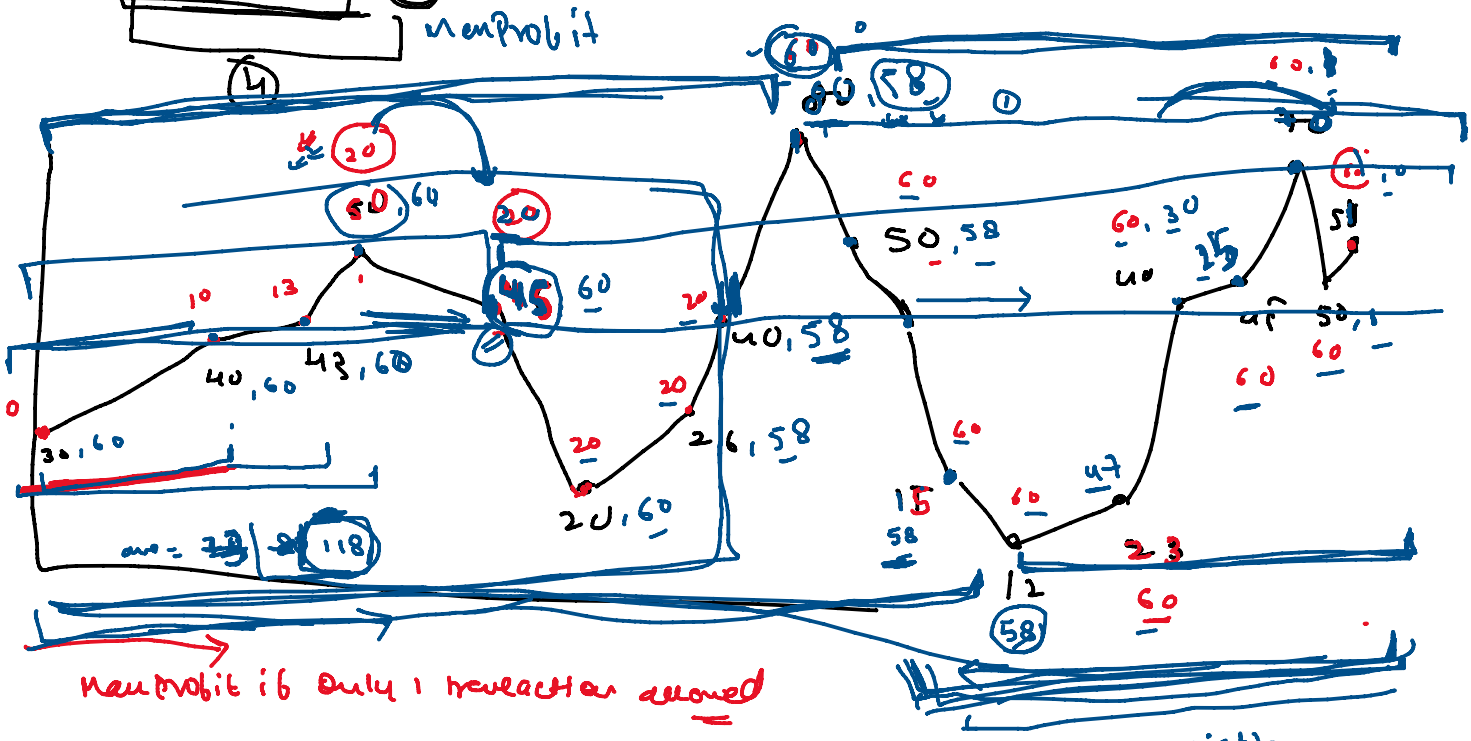
Buy and sell stocks if 2 transactions Allowed

Easy — Only 1 transaction

Buy and sell

3, 3, 5, 1, 0, 3, 1, 4

2 transact



non variable, max

index

= n

0, 1, 2, 3 ---- n

Step-1 vector<int> dp

dp[i]

max profit if 1 transaction allow from 0 - ith day

Step-2 → vector<int> dp2 → Max Profit if 1 transaction allowed from i - n

Final ans → max (Step1 + Step2) ;

B B

B S B S B S

max

max

```
int maxProfit(vector<int>& prices) {
        int n=prices.size();
        vector<int>profit(n,0);
        int i;
        int mx=prices[n-1];
        for(i=n-2;i>=0;i--)
        {
            profit[i]=max(profit[i+1],mx-prices[i]);
            mx=max(mx,prices[i]);
        }
        int mn=prices[0];
        for(i=1;i<n;i++)
        {
            profit[i]=max(profit[i-1],profit[i]+prices[i]-mn);
            mn=min(mn,prices[i]);
        }
        return profit[n-1];
    }
```

right to left

left to right

Profit[i] → max profit

n-i

left to right

Profit2 i

Profit[i] = Profit2[i]

Profit[i] = max profit from i to n

min sale

Sell

left to right + (right to left)

→ 1

→ 2 ⟶ ⓐ

→ ⟸ Transaction Fee

→ ∞ transaction

→ K transaction allowed

K Transaction Allowed

Price

N

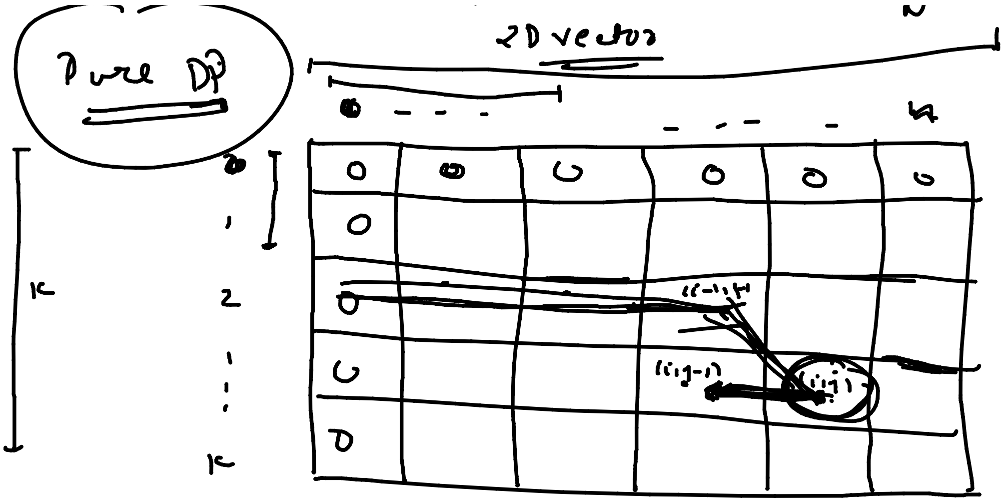
2D vector

Pure DP

0 - - - - n

K

Pure DP



0 - - - n

k

i^{th} day

(i-1) - - - (i)

Max profit you can achieve if

i^{th} transactions are done and j days are performed

(k, n)

(i, j)

(i, j)

0

j^{th} day → 0 transaction

j^{th} day → buy

Buy ✗

No sell

j^{th} → sell, buy ✗

(i, j)

sell ✗

sell

(i, j-1)

(0 - j-1)

i transaction

(j-1) days

finish

j ← (j-1) (j-2) - - - 0

1 transaction

(i-1) transaction

(j-1) days

(0 - j-1)

j day sell

j
0
1
2
!
(j-1) → 1 transaction j sell

(i-1) [0 - (j-2)] days

i

j

(i-1) 0 - (j-1)

(i-1)

(j-1) - j

(0 - j-2)
(i-1)

j-2 → j
(1)

sell
→ (i-1, j-1) +
→ (i-1, j-2) + Profit (Price(j) - Price(j-2))
.
.

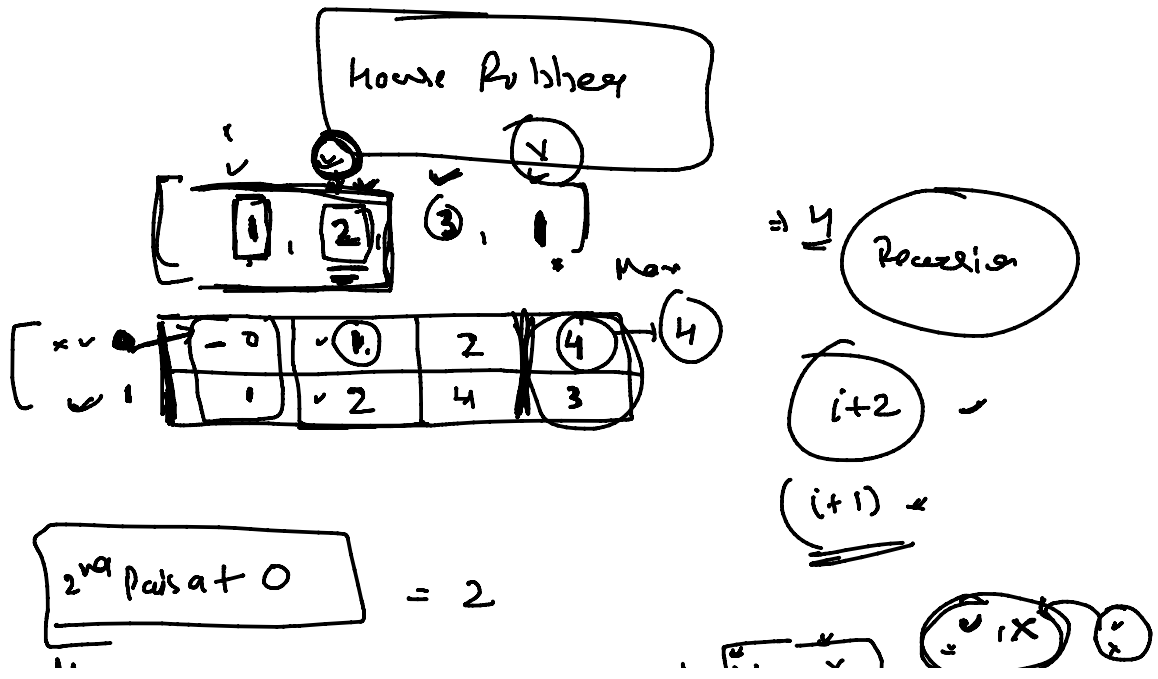
(i-1, j-1)

```
Vector < vector < int > > dp(k+1, vector<int>(n));

for (i=0; i<n; i++)
  dp[0][i] = 0;

for (i=0; i<k; i++)
  dp[i][0] = 0;

for (i=1; i<=k; i++)
{ for (j=1; j<=n; j++)
  { int mn = dp[i][j-1];        (0 -- j-1)
    for (z=0; z<j; z++)
    { mx = max(mn, dp[i-1][z] + Prices[i] - Prices[z]);
    }
  dp[i][j] = mx;
  }
}
```



2... pairs at 0 = 2

Find

2, 7, 9, 3, 1 → Transaction fee

✓, X → ✓ X

include exclusion

0	0	2	7	11	[illegible]
1	2	7	11	10	12

max (dp[n][0], dp[n][1])

9 + dp[i-1][0]

dp[i]

0 - i

```
vector<vector<int>> dp (n+1, vector<int> ( 2,0 ));

dp[0][0] = 0;

dp[0][1] = nums[0];

for (i=1; i<n; i++)
{ dp[i][0] = max ( dp[i-1][0], dp[i-1][1] );
    dp[i][1] = nums[i] + dp[i-1][0];
}

return max ( dp[n][0], dp[n][1] );
```

Fri

Sat