23. Best Time to Buy and Sell Stock

121. Best Time to Buy and Sell Stock

Easy 24.2K 756

Companies

You are given an array `prices` where `prices[i]` is the price of a given stock on the i^{th} day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock.

Return *the maximum profit you can achieve from this transaction*. If you cannot achieve any profit, return `0`.

Example 1:

```
Input: prices = [7,1,5,3,6,4]
Output: 5
Explanation: Buy on day 2 (price = 1) and sell on day 5
(price = 6), profit = 6-1 = 5.
Note that buying on day 2 and selling on day 1 is not
allowed because you must buy before you sell.
```

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

→ You need to decide a day when you buy a stock and sell a stock on other day

→ You need to maximize the profit

for e.g

1st day Buy → 1 RS

3rd day Sell → 6 RS

5 RS

Remember inorder to sell a stock you need to buy it.

If you are selling on i^{th} day you buy on the min price from 1st → (i−1) day

7 1 5 3 6 4

→ You won't sell on the same day so you start at 1st index.

7 1 5 3 6 4

$5-1=4$ (Profit)

$3-1=2$

$6-1=5$ for the 2nd day this is the max profit

$4-1=3$

So for every guy keep track of the minimal on the left my job will be done.

Pseudo Code:

```
mini = a[0], profit = 0
for(i=1; i<n; i++)
{
    cost = a[i] - mini;
    profit = max(profit, cost);
    mini = min(mini, a[i]); // this is for the next
                            iteration you can keep a track
                            of the minimum so while moving
                            you keep all the track of all the
}
print(profit)
```

You keep all the track of all the minimals of a of i so that for the rest of a[i] i'll be having the minimal.

```
class Solution {
public:
    int maxProfit(vector<int>& prices) {
        int mini = prices[0];
        int maxProfit = 0;
        int n = prices.size();
        for(int i = 1; i < n; i++){
            int cost = prices[i] - mini;
            maxProfit = max(maxProfit, cost);
            mini = min(mini, prices[i]);
        }
        return maxProfit;
    }
};
```