

# Leetcode 121. Best Time to Buy and Sell Stock

2024.02.25

원루빈



# Problem Description - Best Time to Buy and Sell Stock

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.

## Example 2:

**Input:** prices = [7,6,4,3,1]

**Output:** 0

### Explanation:

In this case, no transactions are done and the max profit = 0.



## Intuition

prices = [3, 8, 1, 4, 7, 5]

Max\_profit: 6

Buy : -1

Sell: + 7

→ Total profit:  $-1 + 7 = 6$

profit



- Buy at the lowest price & Sell at the highest price
- Buy -> sell order must be maintained.

1

Brute Force Solution → Finding every single combination

	3	8	1	4	7	5
3						
8						
1						
4						
7						
5						

1

Brute Force Solution → Finding every single combination

Row3: Buy at 3 →

	3	8	1	4	7	5
3		5	-2	1	4	2
8						
1						
4						
7						
5						

1

Brute Force Solution → Finding every single combination

Buy at 3 →

	3	8	1	4	7	5
3		5	-2	1	4	2
8						
1						
4						
7						
5						

Calculate the profit : (Sell - Buy)

1

Brute Force Solution → Finding every single combination

Buy at 8 →

	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1						
4						
7						
5						

Calculate the profit : (Sell - Buy)

1

Brute Force Solution → Finding every single combination

Continue on...

	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

Calculate the profit : (Sell - Buy)



1

Brute Force Solution → Finding every single combination

We found the max profit!

	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

Calculate the profit : (Sell - Buy)

## 2

## Greedy Algorithm (Optimized Solution)



	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

Profit: **SELL** - **BUY** = 8 - 3 = + 5

## 2

## Greedy Algorithm (Optimized Solution)



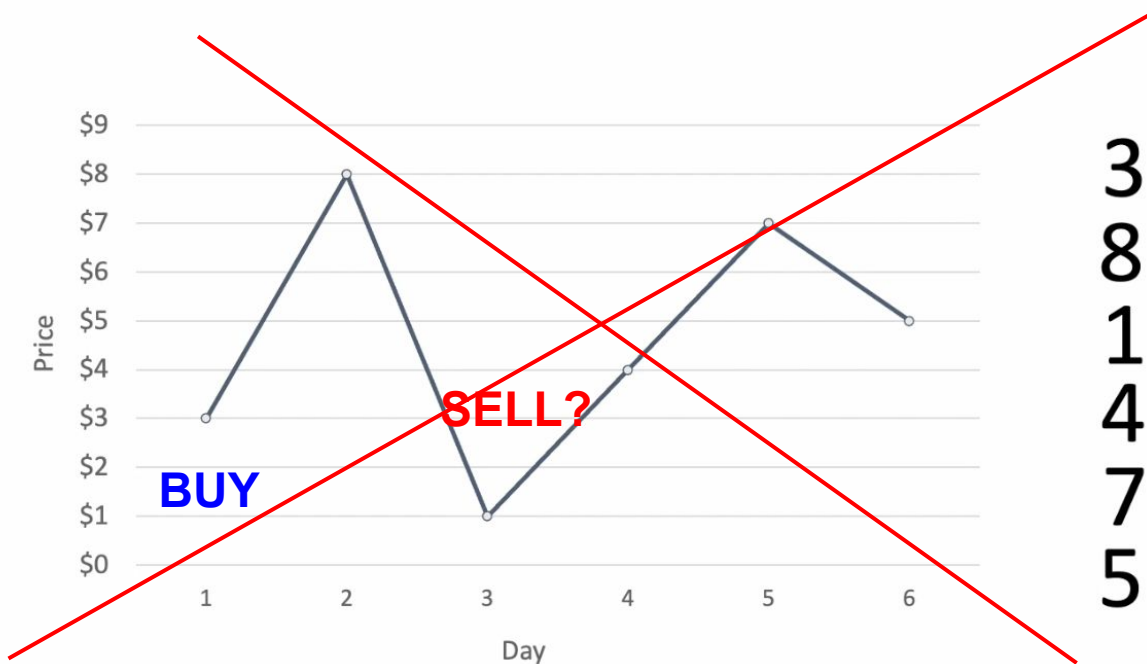
	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

Profit: SELL - BUY = 1 - 3 = -2

# 2

## Greedy Algorithm (Optimized Solution)



	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

Profit: SELL - BUY = 1 - 3 = -2

## 2

## Greedy Algorithm (Optimized Solution)



	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

Profit: SELL - BUY = 7 - 3 = 4

## 2

## Greedy Algorithm (Optimized Solution)



	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

[Key point] Sell should be higher than Buy to obtain a profit!

# 2

## Greedy Algorithm (Optimized Solution)



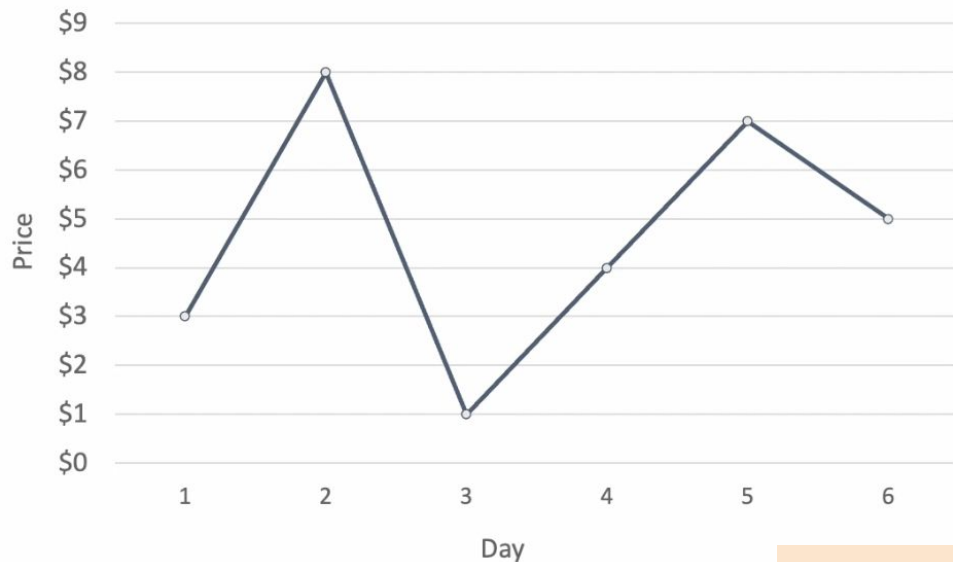
	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

[Key point] Sell should be higher than Buy to obtain a profit!

## 2

## Greedy Algorithm (Optimized Solution)



prices = [3, 8, 1, 4, 7, 5]

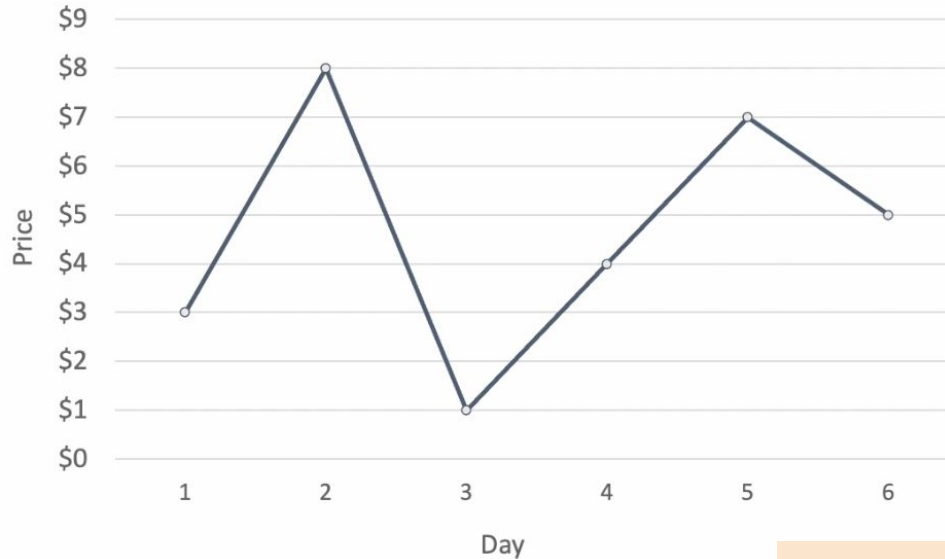
	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

**[Key point] Sell should be higher than Buy to obtain a profit!**



# 2

## Greedy Algorithm (Optimized Solution)



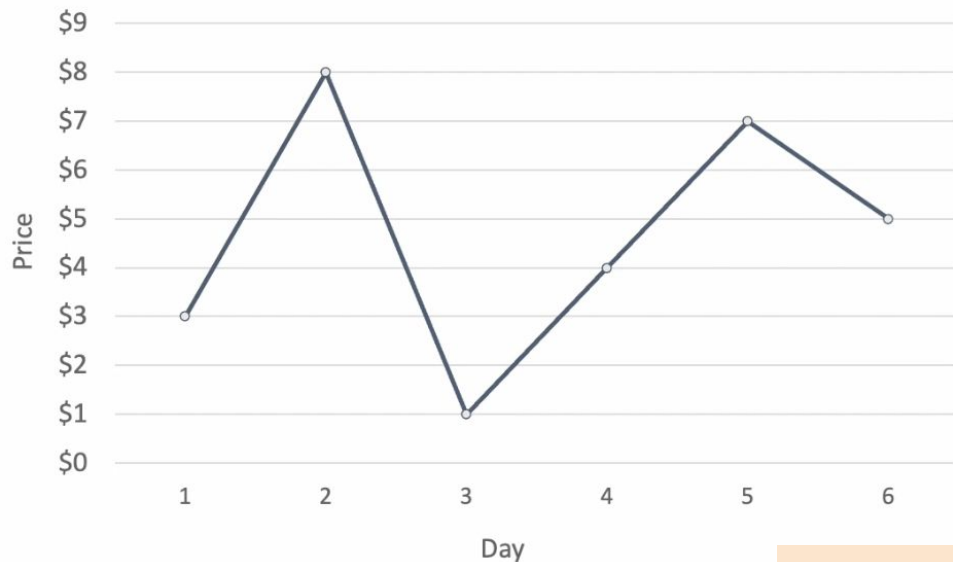
	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

**[Key point] Sell should be higher than Buy to obtain a profit!**

# 2

## Greedy Algorithm (Optimized Solution)



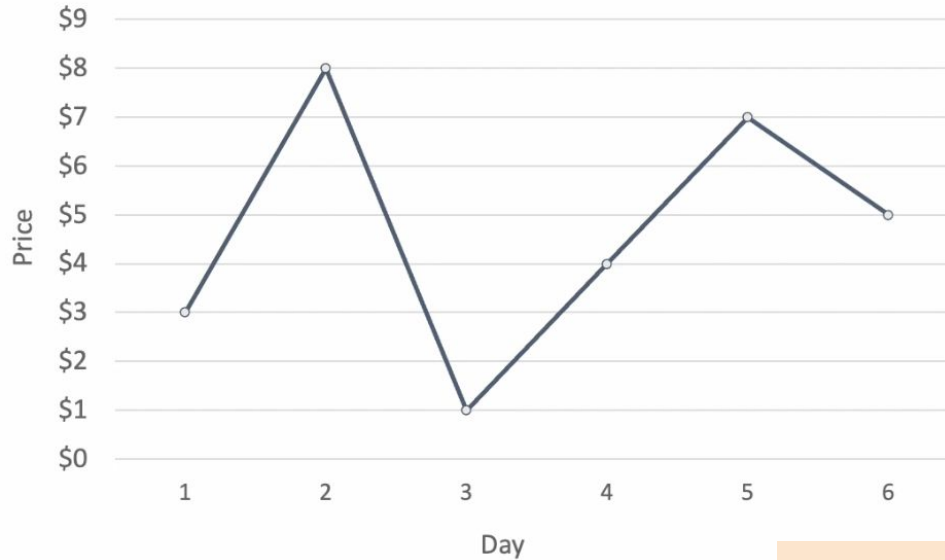
prices = [3, 8, 1, 4, 7, 5]

	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

**[Key point] Sell should be higher than Buy to obtain a profit!**

# 2

## Greedy Algorithm (Optimized Solution)



	3	8	1	4	7	5
3		5	-2	1	4	2
8			-7	-4	-1	-3
1				3	6	4
4					3	1
7						-2
5						

prices = [3, 8, 1, 4, 7, 5]

We only have to compare 5 times in total =>  $O(n)$  time complexity



## Code Solution - Greedy Algorithm

```
def maxProfit(self, prices) -> int: # greedy algorithm
    buy = prices[0] # min val
    profit = 0

    for sell in prices[1:]:
        if sell > buy: # if profit is positive
            profit = max(profit, sell - buy)
        else:
            buy = sell
    return profit
```



## Code Solution - Using two pointers

```
def maxProfit_two_pointers(self, prices) -> int:
    l, r = 0, 1
    profit = 0
    while r < len(prices):
        if prices[l] < prices[r]:
            profit = max(profit, prices[r] - prices[l])
        else:
            l = r # instead of l+=1 since we want the left to go to the next smallest value.
        r += 1

    return profit
```

감사합니다!

THANK YOU