

## Leetcode Problem 2. (Easy)

### 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  $i^{\text{th}}$  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.

#### Constraints:

- $1 \leq \text{prices.length} \leq 10^5$
- $0 \leq \text{prices}[i] \leq 10^4$

Link: <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

```
class Solution {
    public int maxProfit(int[] prices) {

        int minPrice = Integer.MAX_VALUE;
        int maxProfit = 0;

        for (int i = 0; i < prices.length; i++) {
            if (prices[i] < minPrice) {
                minPrice = prices[i];
            } else if (prices[i] - minPrice > maxProfit) {
                maxProfit = prices[i] - minPrice;
            }
        }
    }
}
```

```
        return maxProfit;
    }
}
```

LeetCode

< Problem List >

Premium

🕒 0

👤

Description Editorial Solutions (11K) Submissions

Accepted

Next question

122. Best Time to Buy and Sell Stock II

More challenges

53. Maximum Subarray 122. Best Time to Buy and Sell Stock II

123. Best Time to Buy and Sell Stock III

All statuses All languages

Accepted a few seconds ago Java

Sakib Rahman

Apr 28, 2023 13:54

Details + Solution

Java

Runtime 2 ms Beats 92.51% Memory 59.3 MB Beats 41.60%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags 0/5

class Solution {  
 public int maxProfit(int[] prices) {  
  
 int minPrice = Integer.MAX\_VALUE;  
 int maxProfit = 0;  
  
 for (int i = 0; i < prices.length; i++) {  
 if (prices[i] < minPrice) {  
 minPrice = prices[i];  
 } else if (prices[i] - minPrice > maxProfit) {  
 maxProfit = prices[i] - minPrice;  
 }  
 }  
 }  
}

Console

Run Submit