Problem List

Run Submit

Description | Accepted | Editorial | Solutions | Submissions

121. Best Time to Buy and Sell Stock

Solved

Easy | Topics | 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.
```

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 <= prices.length <= 10^5`
- `0 <= prices[i] <= 10^4`

31K | 360

00:10:25 0 Premium

Code

Java Auto

```java
class Solution {
    public int maxProfit(int[] prices) {
        if (prices == null || prices.length <= 1) {
            return 0; // If there are no prices or only one price, no profit can be made
        }

        int buy = prices[0]; // Initial buy price
        int maxProfit = 0; // Initialize maximum profit

        // Iterate through the prices to find the maximum profit
        for (int i = 1; i < prices.length; i++) {
            if (prices[i] < buy) {
                buy = prices[i]; // Update buy price to the lowest possible price found so far
            } else {
                int currentProfit = prices[i] - buy; // Calculate current profit if we sell at current price
                if (currentProfit > maxProfit) {
                    maxProfit = currentProfit; // Update maximum profit if current profit is higher
                }
            }
        }

        return maxProfit;
    }
```

Saved Ln 1, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

prices =

[7,1,5,3,6,4]

< Best Time to Sell and Buy Stock >

Medium

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.

Remember > you must buy before you sell.

Solution >

```
If(prices == null || prices.length <= 1){
    return 0; //If there is no price or only one
}                price no profit can be made
```

i>=0

```
int buy = prices[0];        //initial buy Price
int maxProfit = 0;          // Initialize max Profits

// iterate through the Prices to find max profit
for(int i = 1; i< prices.length; i++){
    if( Prices[i] < buy ){
        buy = price[i]; // update buy Price to the lowest
        }                   possible price found so far
    else {
    int currentProfit = profit[i]-buy; // calculate current
    Profit if we sell at current Price
    if(Curren!Profit > maxProfit){
    maxProfit = currentProfit; // updat maximum profit
    } if current profil is higher
  }
}
return maxProfit;
}
```

idx1

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted

Azan imtiaz submitted at Jul 24, 2024 09:55

Editorial | Solution

Runtime

2 ms | Beats 70.16%

Analyze Complexity

Memory

61.56 MB | Beats 53.66%

Analyze Complexity

Code | Java

```
class Solution {
    public int maxProfit(int[] prices) {
        if (prices == null || prices.length <= 1) {
            return 0; // If there are no prices or only one price, no profit can
```