



[Click on the logo to find **Problem Statement**]

## Intuition

---

The code aims to find the maximum profit that can be obtained from a given list of stock prices. It employs a simple and efficient one-pass algorithm. The variables `buy` and `sell` represent indices, and the algorithm iterates through the prices, tracking the potential buying and selling points. When a price at the current `sell` index is higher than the price at the current `buy` index, it calculates the profit and updates the `max_profit` accordingly. If the current `sell` price is not greater, it updates the `buy` index to the current `sell` index. This way, it effectively identifies the local minimum and maximum points in the price trend, providing the maximum profit achievable in a single transaction. The time complexity is  $O(n)$  as it processes each price once, and the space complexity is  $O(1)$  as it uses a constant amount of additional memory.

## Approach

---

### 1. Initialization:

- Initialize `buy` to 0 and `sell` to 1.
- Set `max_profit` to 0.

### 2. Iteration:

- While the `sell` index is within the bounds of the prices list:
  - Check if the price at the `buy` index is less than the price at the `sell` index.
    - If true, calculate the profit as the difference between the prices and update `max_profit` if the current profit is greater.
    - If false, update the `buy` index to the current `sell` index.

### 3. Move to the Next Index:

- Increment the `sell` index to move to the next potential selling point.

### 4. Result:

- The final `max_profit` represents the maximum profit that can be obtained from a single transaction.

# Complexity

---

- Time complexity:  $O(n)$
- Space complexity:  $O(1)$

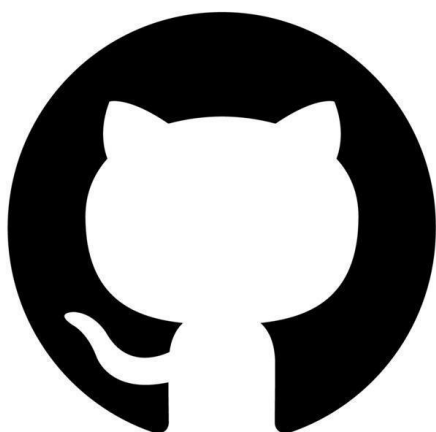
# Code

---

```
class Solution:
    def maxProfit(self, prices: List[int]) -> int:
        buy = 0
        sell = 1
        max_profit = 0
        while sell < len(prices):
            if prices[buy] < prices[sell]:
                profit = prices[sell] - prices[buy]
                max_profit = max(max_profit, profit)
            else:
                buy = sell
            sell += 1

        return max_profit
```

If you want to see more solutions to coding problems, you can visit:



# GitHub