LeetCode – Best Time to Buy and Sell Stock III (Java)

Say you have an array for which the ith element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete at most two transactions.

Note:
A transaction is a buy & a sell. You may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

Analysis

Comparing to I and II, III limits the number of transactions to 2. This can be solve by "devide and conquer". We use left[i] to track the maximum profit for transactions before i, and use right[i] to track the maximum profit for transactions after i. You can use the following example to understand the Java solution:

```
Prices: 1 4 5 7 6 3 2 9
left = [0, 3, 4, 6, 6, 6, 6, 8]
right= [8, 7, 7, 7, 7, 7, 7, 0]
```

The maximum profit = 13

Java Solution

```
public int maxProfit(int[] prices) {
        if (prices == null || prices.length < 2) {
                return 0;
        }

        //highest profit in 0 ... i
        int[] left = new int[prices.length];
        int[] right = new int[prices.length];

        // DP from left to right
        left[0] = 0;
        int min = prices[0];
        for (int i = 1; i < prices.length; i++) {
                min = Math.min(min, prices[i]);
                left[i] = Math.max(left[i - 1], prices[i] - min);
        }

        // DP from right to left
```

```
        right[prices.length - 1] = 0;
        int max = prices[prices.length - 1];
        for (int i = prices.length - 2; i >= 0; i--) {
                max = Math.max(max, prices[i]);
                right[i] = Math.max(right[i + 1], max - prices[i]);
        }

        int profit = 0;
        for (int i = 0; i < prices.length; i++) {
                profit = Math.max(profit, left[i] + right[i]);
        }

        return profit;
}
```

Related Posts:

1. **LeetCode – Best Time to Buy and Sell Stock (Java)**
2. **LeetCode – Best Time to Buy and Sell Stock II (Java)**
3. **LeetCode – Best Time to Buy and Sell Stock IV (Java)**
4. **LeetCode – Maximum Product Subarray (Java)**

Category >> Algorithms

If you want someone to read your code, please put the code inside <pre><code> and </code></pre> tags. For example:

```
<pre><code>
String foo = "bar";
</code></pre>
```

min = Math.min(min, prices[i]);
this has to e after finding left[i].

Reply • Share ›

Holden • 3 years ago

Could you please explain how you get this array?
right= [8, 7, 7, 7, 7, 7, 7, 0]

Reply • Share ›

Holden • 3 years ago

How can we return the indexes as well?

Reply • Share ›