

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(nk) time | O(n) space
4 function maxProfitWithKTransactions(prices, k) {
5   if (!prices.length) return 0;
6   const evenProfits = new Array(prices.length).fill(0);
7   const oddProfits = new Array(prices.length).fill(0);
8   for (let t = 1; t < k + 1; t++) {
9     let maxThusFar = -Infinity;
10    let currentProfits, previousProfits;
11    if (t % 2 === 1) {
12      currentProfits = oddProfits;
13      previousProfits = evenProfits;
14    } else {
15      currentProfits = evenProfits;
16      previousProfits = oddProfits;
17    }
18    for (let d = 1; d < prices.length; d++) {
19      maxThusFar = Math.max(maxThusFar, previousProfits[d - 1] - prices[d - 1]);
20      currentProfits[d] = Math.max(currentProfits[d - 1], maxThusFar + prices[d] - prices[d - 1]);
21    }
22  }
23   return k % 2 === 0 ? evenProfits[prices.length - 1] : oddProfits[prices.length - 1];
24 }
25
26 exports.maxProfitWithKTransactions = maxProfitWithKTransactions;
27
```

Solution 1

Solution 2

Solution 3

```
1 function maxProfitWithKTransactions(prices, k) {
2   // Write your code here.
3 }
4
5 // Do not edit the line below.
6 exports.maxProfitWithKTransactions = maxProfitWithKTransactions;
7
```

Our Tests

Custom Output

Submit Code

```
1 // Test Case 1: Expected: 0
2 // Input: prices = [], k = 0
3
4 // Test Case 2: Expected: 0
5 // Input: prices = [1, 2, 3, 4, 5], k = 0
6
7 // Test Case 3: Expected: 4
8 // Input: prices = [1, 2, 3, 4, 5], k = 1
9
10 // Test Case 4: Expected: 6
11 // Input: prices = [1, 2, 3, 4, 5], k = 2
12
```

```
100 # Test Case 100 - Timeout (100)
101 # Run expect program with timeout of 100 seconds (100, 100, 100)
102 #
103 # Test Case 101 - Timeout (101)
104 # Run expect program with timeout of 100 seconds (100, 100, 100)
105 #
106 # Test Case 102 - Timeout (102)
107 # Run expect program with timeout of 100 seconds (100, 100, 100)
108 #
109 # Test Case 103 - Timeout (103)
110 # Run expect program with timeout of 100 seconds (100, 100, 100)
111 #
```

Run or submit code when you're ready.