

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(nk) time | O(nk) space
5     public static int maxProfitWithKTransactions(int[] prices, int k) {
6         if (prices.length == 0) {
7             return 0;
8         }
9         int[][] profits = new int[k + 1][prices.length];
10        for (int t = 1; t < k + 1; t++) {
11            int maxThusFar = Integer.MIN_VALUE;
12            for (int d = 1; d < prices.length; d++) {
13                maxThusFar = Math.max(maxThusFar, profits[t - 1][d - 1] - price
14                profits[t][d] = Math.max(profits[t][d - 1], maxThusFar + price
15            }
16        }
17        return profits[k][prices.length - 1];
18    }
19 }
20
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     public static int maxProfitWithKTransactions(int[] prices, int k) {
3         // Write your code here.
4         return -1;
5     }
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 class Program {
2     // Write your code here.
3     public static int maxProfitWithKTransactions(int[] prices, int k) {
4         // Write your code here.
5     }
6 }
```

```
10 #Print
11 #Print TestPasses: 0
12 #Print Test = 100
13 #Print assert(assertTrue(assertTestPasses, 10) == 1)
14 }
15
16 #Print
17 #Print TestPasses: 0
18 #Print Test = 10, 100
19 #Print assert(assertTrue(assertTestPasses, 10) == 1)
20 }
21
22 #Print
23 #Print TestPasses: 0
24 #Print Test = 10, 100
```

Run or submit code when you're ready.