

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 def maxProfitWithKTransactions(prices, k):
5     if not len(prices):
6         return 0
7     evenProfits = [0 for d in prices]
8     oddProfits = [0 for d in prices]
9     for t in range(1, k + 1):
10         maxThusFar = float("-inf")
11         if t % 2 == 1:
12             currentProfits = oddProfits
13             previousProfits = evenProfits
14         else:
15             currentProfits = evenProfits
16             previousProfits = oddProfits
17         for d in range(1, len(prices)):
18             maxThusFar = max(maxThusFar, previousProfits[d - 1] - prices[d - 1])
19             currentProfits[d] = max(currentProfits[d - 1], maxThusFar)
20     return evenProfits[-1] if k % 2 == 0 else oddProfits[-1]
21
```

Solution 1

Solution 2

Solution 3

```
1 def maxProfitWithKTransactions(prices, k):
2     # Write your code here.
3     pass
4
```

Our Tests

Custom Output

Submit Code

1 prices = [1, 2, 3, 4, 5]

2 k = 2

3

4 Expected: 6

5 Actual: 6

6

7 prices = [2, 1, 2, 1, 0, 1, 1, 2]

8 k = 3

9 Expected: 4

10 Actual: 4

11

12 prices = [3, 2, 6, 4, 1]

13 k = 1

14 Expected: 2

15 Actual: 2

16

17 prices = [1, 2, 3, 4, 5]

18 k = 0

19 Expected: 0

20 Actual: 0

21

```
10 # Add a new row to the DataFrame
11 df = df.append({'name': 'Alice', 'age': 25, 'height': 160})
12
13 # Add a new row to the DataFrame
14 df = df.append({'name': 'Bob', 'age': 30, 'height': 180})
15
16 # Add a new row to the DataFrame
17 df = df.append({'name': 'Charlie', 'age': 35, 'height': 190})
18
19 # Add a new row to the DataFrame
20 df = df.append({'name': 'David', 'age': 40, 'height': 200})
21
22 # Add a new row to the DataFrame
23 df = df.append({'name': 'Eve', 'age': 45, 'height': 210})
```

Run or submit code when you're ready.