

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 import "math"
6
7 // O(nk) time | O(nk) space
8 func MaxProfitWithKTransactions(prices []int, k int) int {
9     if len(prices) == 0 {
10         return 0
11     }
12     profits := make([][]int, k+1)
13     for i := range profits {
14         profits[i] = make([]int, len(prices))
15     }
16     for t := 1; t < k+1; t++ {
17         maxThusFar := math.MinInt32
18         for d := 1; d < len(prices); d++ {
19             maxThusFar = max(maxThusFar, profits[t-1][d-1]-prices[d-1])
20             profits[t][d] = max(profits[t][d-1], maxThusFar+prices[d])
21         }
22     }
23     return profits[k][len(prices)-1]
24 }
25
26 func max(arg int, rest ...int) int {
27     curr := arg
28     for _, num := range rest {
29         if curr < num {
30             curr = num
31         }
32     }
33     return curr
34 }
```

Solution 1Solution 2Solution 3

```
1 package main
2
3 func MaxProfitWithKTransactions(prices []int, k int) int {
4     // Write your code here.
5     return -1
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 import "math"
6
7 // O(nk) time | O(nk) space
8 func MaxProfitWithKTransactions(prices []int, k int) int {
9     if len(prices) == 0 {
10         return 0
11     }
12     profits := make([][]int, k+1)
13     for i := range profits {
14         profits[i] = make([]int, len(prices))
15     }
16     for t := 1; t < k+1; t++ {
17         maxThusFar := math.MinInt32
18         for d := 1; d < len(prices); d++ {
19             maxThusFar = max(maxThusFar, profits[t-1][d-1]-prices[d-1])
20             profits[t][d] = max(profits[t][d-1], maxThusFar+prices[d])
21         }
22     }
23     return profits[k][len(prices)-1]
24 }
25
26 func max(arg int, rest ...int) int {
27     curr := arg
28     for _, num := range rest {
29         if curr < num {
30             curr = num
31         }
32     }
33     return curr
34 }
```

```
1 package main
2
3 func MaxProfitWithKTransactions(prices []int, k int) int {
4     // Write your code here.
5     return -1
6 }
7
```

```
10 # Returns Report, a BeautifulSoup BeautifulSoup object
11
12 # Run in Threadpool Testpool Testpool 2
13 # Returns Report, a BeautifulSoup BeautifulSoup object
14
15 #
16
17 # Run in Threadpool Testpool Testpool 2
18 # Returns Report, a BeautifulSoup BeautifulSoup object
19
20 #
21
22 # Run in Threadpool Testpool Testpool 2
23 # Returns Report, a BeautifulSoup BeautifulSoup object
24
25 #
26
```

Run or submit code when you're ready.