

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 using System;
4 public class Program {
5     // O(nk) time | O(n) space
6     public static int MaxProfitWithKTransactions(int[] prices, int k) {
7         if (prices.Length == 0) {
8             return 0;
9         }
10        int[] evenProfits = new int[prices.Length];
11        int[] oddProfits = new int[prices.Length];
12        for (int i = 0; i < prices.Length; i++) {
13            evenProfits[i] = 0;
14            oddProfits[i] = 0;
15        }
16        for (int t = 1; t < k + 1; t++) {
17            int maxThusFar = Int32.MinValue;
18            int[] currentProfits = new int[prices.Length];
19            int[] previousProfits = new int[prices.Length];
20            if (t % 2 == 1) {
21                currentProfits = oddProfits;
22                previousProfits = evenProfits;
23            } else {
24                currentProfits = evenProfits;
25                previousProfits = oddProfits;
26            }
27            for (int d = 1; d < prices.Length; d++) {
28                maxThusFar =
29                    Math.Max(maxThusFar,
30                        previousProfits[d - 1] - prices[d - 1]);
31                currentProfits[d] = Math.Max(currentProfits[d - 1],
32                    maxThusFar + prices[d]);
33            }
34        }
35        return k % 2 == 0 ? evenProfits[prices.Length - 1] : oddProfits[prices.Length - 1];
36    }
37 }
38
```

