Our Solution(s) Run Code

Solution 1 Solution 2

```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using System;
public class Program {
  // O(nk) time | O(n) space
  public static int MaxProfitWithKTransactions(int[] prices, int k) {
    if (prices.Length == 0) {
      return 0;
    }
    int[] evenProfits = new int[prices.Length];
    int[] oddProfits = new int[prices.Length];
    for (int i = 0; i < prices.Length; i++) {
      evenProfits[i] = 0;
      oddProfits[i] = 0;
    }
    for (int t = 1; t < k + 1; t++) {
      int maxThusFar = Int32.MinValue;
      int[] currentProfits = new int[prices.Length];
      int[] previousProfits = new int[prices.Length];
      if (t % 2 == 1) {
        currentProfits = oddProfits;
        previousProfits = evenProfits;
      } else {
        currentProfits = evenProfits;
        previousProfits = oddProfits;
      }
      for (int d = 1; d < prices.Length; d++) {
        maxThusFar =
          Math.Max(maxThusFar,
            previousProfits[d - 1] - prices[d - 1]);
        currentProfits[d] = Math.Max(currentProfits[d - 1],
            maxThusFar + prices[d]);
      }
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
public class Program {
  public static int MaxProfitWithKTransactions(int[] prices, int k) {
    // Write your code here.
    return -1;
  }
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.