Our Solution(s) Run Code

Solution 1 Solution 2

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

package main

import "math"

// O(nk) time | O(nk) space
func MaxProfitWithKTransactions(prices []int, k int) int {
  if len(prices) == 0 {
    return 0
  }
  profits := make([][]int, k+1)
  for i := range profits {
    profits[i] = make([]int, len(prices))
  }
  for t := 1; t < k+1; t++ {
    maxThusFar := math.MinInt32
    for d := 1; d < len(prices); d++ {
      maxThusFar = max(maxThusFar, profits[t-1][d-1]-prices[d-1])
      profits[t][d] = max(profits[t][d-1], maxThusFar+prices[d])
    }
  }
  return profits[k][len(prices)-1]
}

func max(arg int, rest ...int) int {
  curr := arg
  for _, num := range rest {
    if curr < num {
      curr = num
    }
  }
  return curr
}
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```go
package main

func MaxProfitWithKTransactions(prices []int, k int) int {
  // Write your code here.
  return -1
}

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.