00:00:00 Swift Sublime AlgoExpert **Quad Layout** 12px Monokai

Our Solution(s) Video Explanation Scratchpad Prompt

```
_{\rm 1} \, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
     class Program {
          // O(nk) time | O(nk) space
           \label{lem:func_maxProfitWithKTransactions} \begin{subarray}{ll} func &maxProfitWithKTransactions($\_$ prices: [Int], $\_$ k: Int) $$\rightarrow$ Int { } \end{subarray}
               if prices.count == 0 {
                    return 0
10
               var profits = [[Int]]()
11
12
               for _ in stride(from: 0, through: k, by: 1) {
13
                    let row = Array(repeating: 0, count: prices.count)
14
                    profits.append(row)
15
16
17
               for transaction in stride(from: 1, through: k, by: 1) {
18
                    var maxProfitThusFar = Int.min
19
20
21
22
                    for day in stride(from: 1, to: prices.count, by: 1) {
   maxProfitThusFar = max(maxProfitThusFar, profits[transaction - 1][day - 1] - prices[day - 1])
                         profits[transaction][day] = max(profits[transaction][day - 1], \ maxProfitThusFar + prices[day])
23
24
25
26
               return profits[k][prices.count - 1]
27
28 }
29
```

Solution 1 Solution 2

Run Code