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

Our Solution(s) Video Explanation Scratchpad Run Code

Solution 1 Solution 2

Prompt

34

```
_{\rm 1} \, // Copyright 0 2020 AlgoExpert, LLC. All rights reserved.
    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];
10
         for (int i = 0; i < prices.length; i++) {</pre>
11
           evenProfits[i] = 0;
12
13
           oddProfits[i] = 0;
14
         for (int t = 1; t < k + 1; t++) {
  int maxThusFar = Integer.MIN_VALUE;</pre>
15
16
17
            int[] currentProfits = new int[prices.length];
18
            int[] previousProfits = new int[prices.length];
19
            if (t % 2 == 1) {
20
21
             currentProfits = oddProfits;
              previousProfits = evenProfits;
22
            } else {
23
             currentProfits = evenProfits;
24
              previousProfits = oddProfits;
25
26
            for (int d = 1; d < prices.length; d++) {</pre>
27
28
29
             maxThusFar = Math.max(maxThusFar, previousProfits[d - 1] - prices[d - 1]);
currentProfits[d] = Math.max(currentProfits[d - 1], maxThusFar + prices[d]);
30
31
         return k % 2 == 0 ? evenProfits[prices.length - 1] : oddProfits[prices.length - 1];
32
33 }
```