

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 // O(nk) time | O(nk) space
4 function maxProfitWithKTransactions(prices, k) {
5   if (!prices.length) return 0;
6   const profits = [];
7   for (let t = 0; t < k + 1; t++) {
8     const row = new Array(prices.length).fill(0);
9     profits.push(row);
10  }
11  for (let t = 1; t < k + 1; t++) {
12    let maxThusFar = -Infinity;
13    for (let d = 1; d < prices.length; d++) {
14      maxThusFar = Math.max(maxThusFar, profits[t - 1][d - 1] - prices[d - 1]);
15      profits[t][d] = Math.max(profits[t][d - 1], maxThusFar + prices[d]);
16    }
17  }
18  return profits[k][prices.length - 1];
19 }
20
21 exports.maxProfitWithKTransactions = maxProfitWithKTransactions;
22
```

