Our Solution(s) Run Code

Solution 1 Solution 2

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

// O(nk) time | O(n) space
function maxProfitWithKTransactions(prices, k) {
  if (!prices.length) return 0;
  const evenProfits = new Array(prices.length).fill(0);
  const oddProfits = new Array(prices.length).fill(0);
  for (let t = 1; t < k + 1; t++) {
    let maxThusFar = -Infinity;
    let currentProfits, previousProfits;
    if (t % 2 === 1) {
      currentProfits = oddProfits;
      previousProfits = evenProfits;
    } else {
      currentProfits = evenProfits;
      previousProfits = oddProfits;
    }
    for (let d = 1; d < prices.length; d++) {
      maxThusFar = Math.max(maxThusFar, previousProfits[d - 1] - pric
      currentProfits[d] = Math.max(currentProfits[d - 1], maxThusFar
    }
  }
  return k % 2 === 0 ? evenProfits[prices.length - 1] : oddProfits[pri
}

exports.maxProfitWithKTransactions = maxProfitWithKTransactions;

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```
function maxProfitWithKTransactions(prices, k) {
  // Write your code here.
}

// Do not edit the line below.
exports.maxProfitWithKTransactions = maxProfitWithKTransactions;

```

Our Tests

Custom Output Submit Code

Run or submit code when you're ready.