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

class Program {
    // O(nk) time | O(nk) space
    func maxProfitWithKTransactions(_ prices: [Int], _ k: Int) -> Int {
        if prices.count == 0 {
            return 0
        }

        var profits = [[Int]]()

        for _ in stride(from: 0, through: k, by: 1) {
            let row = Array(repeating: 0, count: prices.count)
            profits.append(row)
        }

        for transaction in stride(from: 1, through: k, by: 1) {
            var maxProfitThusFar = Int.min

            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])
            }
        }

        return profits[k][prices.count - 1]
    }
}

```