# Problem 188: Best Time to Buy and Sell Stock IV

## Problem Information

**Difficulty:** Hard
**Acceptance Rate:** 48.79%
**Paid Only:** No
**Tags:** Array, Dynamic Programming

## Problem Description

You are given an integer array `prices` where `prices[i]` is the price of a given stock on the `ith` day, and an integer `k`.

Find the maximum profit you can achieve. You may complete at most `k` transactions: i.e. you may buy at most `k` times and sell at most `k` times.

**Note:** You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again).

**Example 1:**

**Input:** k = 2, prices = [2,4,1] **Output:** 2 **Explanation:** Buy on day 1 (price = 2) and sell on day 2 (price = 4), profit = 4-2 = 2.

**Example 2:**

**Input:** k = 2, prices = [3,2,6,5,0,3] **Output:** 7 **Explanation:** Buy on day 2 (price = 2) and sell on day 3 (price = 6), profit = 6-2 = 4. Then buy on day 5 (price = 0) and sell on day 6 (price = 3), profit = 3-0 = 3.

**Constraints:**

* `1 <= k <= 100` * `1 <= prices.length <= 1000` * `0 <= prices[i] <= 1000`

## Code Snippets

### C++:

```
class Solution {
public:
int maxProfit(int k, vector<int>& prices) {

}
};
```

### Java:

```
class Solution {
public int maxProfit(int k, int[] prices) {

}
}
```

### Python3:

```
class Solution:
def maxProfit(self, k: int, prices: List[int]) -> int:
```