CMPS 102 — Fall 2018 — Homework 3

"I have read and agree to the collaboration policy." - ***Kevin Wang***

Solution to Problem 1: Crop Profit

Let $A[1...n]$ be the array of crop prices in the past n days where $n > 2$. $A[i]$ is the crop price on day i.

Algorithm 1 Returns the maximum profit possible in the past n days

```
MAX-PROFIT (A[ ], n):
Initialize index of local minima, min
Initialize index of local maxima, max
Initialize maximum profit, profit = 0
Initialize array index, i = 0
while i < n − 1 do
    while (i < n − 1) and (A[i + 1] ≤ A[i]) do
        i + +
    end while
    if i == n − 1 then
        Break
    end if
    min = i, i + +
    while (i < n) and (A[i] ≥ A[i − 1]) do
        i + +
    end while
    max = (i − 1)
    profit+ = (A[max] − A[min])
end while
```

Claim 1. *This algorithm is optimal and finds the maximum profit.*

Proof. Let $s_i - b_i$ be the profit made from a single sell-buy transaction. The profit of the transactions performed by the algorithm is $(s_1 - b_1) + (s_2 - b_2) + ...$

Assume for the sake of contradiction that $b_1 << s_2$ and the transaction $b_1 – s_2$ is more profitable:

$$\begin{aligned}(s_2 - b_1) - [(s_1 - b_1) + (s_2 - b_2)] &= -(s_1 - b_2) \\ &< 0 \\ \implies (s_2 - b_1) &< (s_1 - b_1) + (s_2 - b_2)\end{aligned}$$

Thus, by contradiction, the algorithm does find the maximum profit. □

The algorithm goes through the length of $A[1...n]$ and each check incrememnts the index by 1. Thus the algorithm completes the check of all n indices in time: $O(n)$.

The algorithm requires an array of size n and stores 3 separate values. Thus the space complexity is: $O(n)$.