CMPS 102 — Fall 2018 – Homework 3

Alyssa Melton

I have read and agree to the collaboration policy.

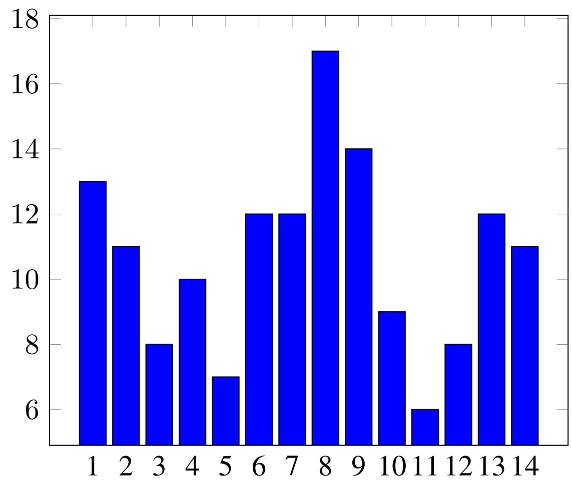
Collaborators: none

Solution to Problem 1

We are given a set of days, 1 to *n*. For each corresponding day, we are given the price of the crop. We want to find the maximum amount of profit that Phillip could have made in these *n* days.

While thinking through different things that might help to solve this problem, I graphed prices of crop over *n* days that were given as an example, namely:

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Price	13	11	8	10	7	12	12	17	14	9	6	8	12	11



We are given that the most profit is gained, for this particular example, is by doing the following:
Buy a cart on day 3, sell the cart on day 4, with a margin of 2;
buy a cart on day 5, sell a cart on day 8, with a margin of 10;
buy a cart on day 11, sell a cart on day 13, with a margin of 6.
Now, comparing these values to the graph above, I noticed that these directly correspond with the peaks and valleys of the graph.

Defintions:
A day's crop price, $price[x]$, is a peak if $price[x] \leq price[x-1]$ and $price[x] \geq A[x+1]$
A day's crop price, $price[x]$, is a valley if $price[x] \geq price[x-1]$ and $price[x] \leq A[x+1]$

Let us assume that the most profit is gained by buying crop at a price valley, and selling crop at a price peak. Now, consider the following algorithm:

$profit = 0.$

$purchaseprice = null.$
$profitmargin = null.price[0] = null.$
$price[n+1] = null.$

for each day i, 1 to n:
    if $price[i-1] \geq price[i]$ and $price[i] \leq price[i+1]$
        if $purchaseprice = null$
            buy crop on day i.
            purchase price = $price[i]$
    else if $price[i-1] \leq price[i]$ and $price[i] \geq price[i+1]$
        if $purchaseprice \neq null$
            sell crop on day i
            $profitmargin = price[i] - purchaseprice$
            $profit = profit + profitmargin$
return *profit*.

Claim 1. *The most profit is gained by buying crop at a price valley, and selling crop at a price peak. (The algorithm is optimal)*

Proof. Assume that the most profit is not gained by buying crop at a price valley and selling crop at a price peak. Then there must be some other algorithm call it *B* that is optimal. Say that *B* and *A* are the same up until day *k*, where *B* either buys or sells at a time that *A* does not.

Case 1: At day *k*, *B* buys at a time that is not a valley.

Call the day which is a valley that *A* buys on j. Then by definition of a valley, j is necessarily cheaper/less expensive than k. Then, no matter where in the future *B* sells, be it the highest priced day after that buy, call it h, $(price(h) - price(j)) > (price(h) - price(k))$, since $k > j$. Thus, *O* made no more profit than *A*.

Case2: At day *k*, *O* sells at a time that is not a peak.

Call the day which is a peak that *A* sells on j. Then by definition of a peak, j is necessarily more valuable/more expensive than k. Then, no matter where in the past *B* bought, be it the lowest priced day before that buy, call it h, $(price(j) - price(h)) > (price(k) - price(h))$, since $j > k$. Thus, *O* made no more profit than *A*.

□

Claim 2. *The above algorithm is $O(n)$.*

Proof. For each day, 1 to *n* we are doing two comparisons, and either buying, selling, or doing nothing. If we call each comparison +1 and each buy or sell +1, then we are doing at most 3 operations every day, for n days. This is at most $3n$ operations, which is $O(n)$. □