BEST TIME TO BUY AND SELL STOCK WITH COOLDOWN

By - Shobhit

Content:

- Problem Statement
- Explanation
- Formula
- Complexities

Problem Statement:(Ref: Leetcode)

Say you have an array for which the *i*th element is the price of a given stock on day *i*.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (ie, buy one and sell one share of the stock multiple times) with the following restrictions:

- You may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).
- After you sell your stock, you cannot buy stock on next day. (ie, cooldown 1 day)

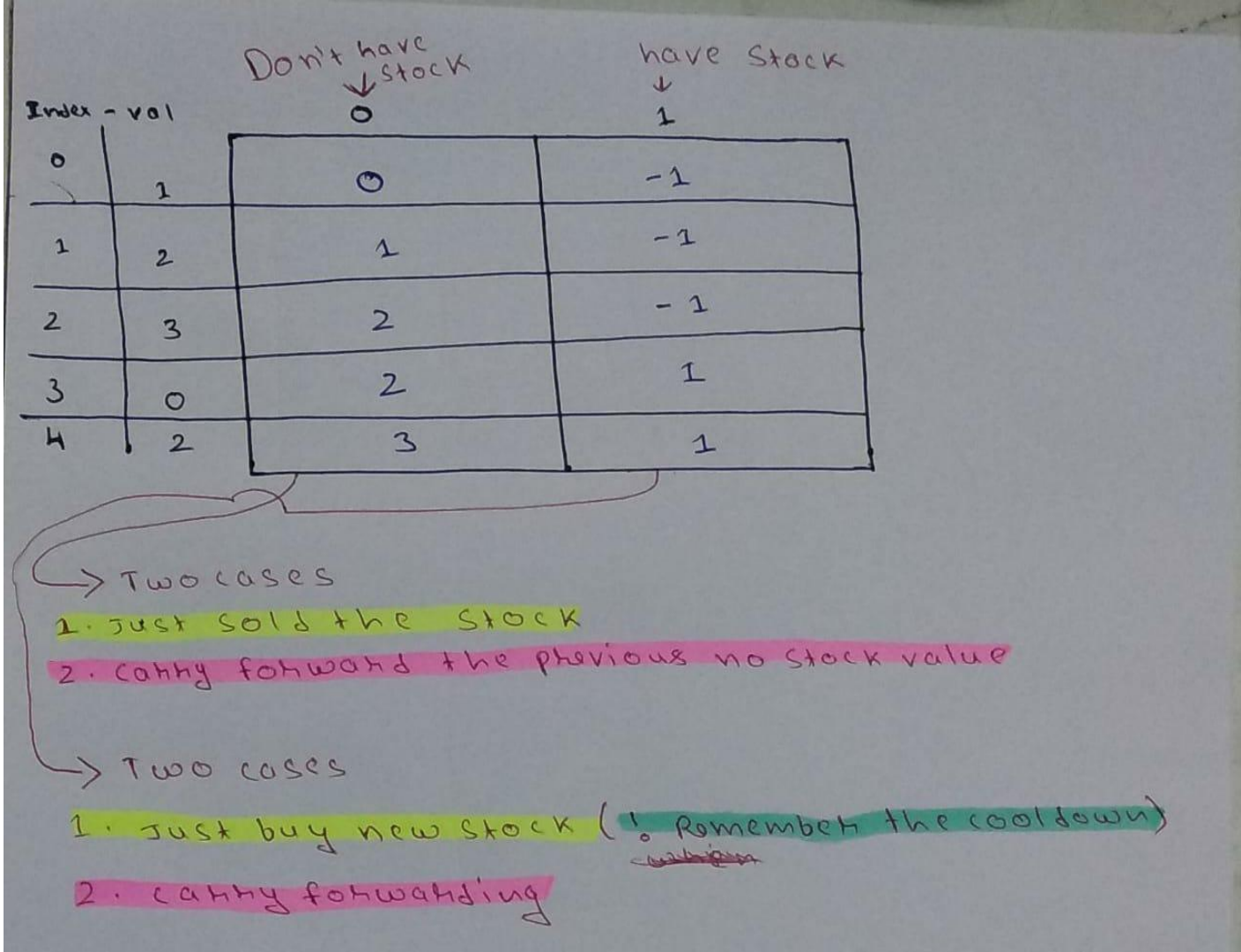
Example:

Input: [1,2,3,0,2]

Output: 3

Explanation: transactions = [buy, sell, cooldown, buy, sell]

Explanation:



formula :

Base cases :

if (length <= 0) return 0

if (length == 2)

if (prices[1] > prices[0]) return prices[1] - prices[0]

else return 0

formula :

dp[i][0] = max (① dp[i-1][1] + prices[i] , ② dp[i-1][0])

dp[i][1] = max (③ dp[i-2][0] - prices[i] , ④ dp[i-1][1])

Complexities:

Time: O(n)

Space: O(n)

Source Code:

https://github.com/shobhit-saini/Leet_Code/blob/master/309.Best_Time_to_Buy_and_Sell_Stock_with_Cooldown.cpp

THANK YOU