SRM Institute of Science and Technology

18CSC204J

Design and Analysis of Algorithms

MINOR PROJECT

MAXIMUM PROFIT BY BUYING AND SELLING A SHARE AT MOST K TIMES

SRMI INSTITUTE OF SCIENCE AND TECHNOLOGY

S.R.M. NAGAR, KATTANKULATHUR -603 203

BONAFIDECERTIFICATE

Register No.______________

Certified to be bonafide record of the work done by ____ of __________________________________, B.tech degree course in the practical 18CSC204J- Design and Analysis of Algorithms in SRM Institute of Science and Technology, Kattankulathur during the academic year 2021-22.

Date: **Lab In charge:**

Submitted for university examination held in __________________

SRM Institute of Science and Technology, Kattankulathur.

School Of Data Science and Business Systems

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC204J

Course Name: Design and Analysis of Algorithm

Title of Experiment	Maximum profit by buying and selling a share at most K times.
Name of the candidate	Challa Avinash
Team Members	Dinesh vadlaputi, abdul Muthallib, Akash Pandit
Register Number	RA2011027010107, RA2011027010121, RA2011027010122, RA20110270101023
Date of Experiment	20-06-2022

Aim: Solving Maximum profit by buying and selling a share at most K times by using dynamic programming

CONTENTS

Problem Definition

Problem Statement: Maximum profit by buying and selling a share at most K times.
In share trading, a buyer buys shares and sells on a future date. Given the stock price of N days, the trader is allowed to make at most K transactions, where a new transaction can only start after the previous transaction is complete. The task is to find out the maximum profit that a share trader could have made.

EUR/USD - 1,35379 - 00:00:00 14 giu (EEST)
EUR/USD (Bid), Ticks, # 300 / 300
Gold, spot - 1,276.820 - 23:00:00 13 giu (CEST)
Gold, spot (Bid), 1 minute, # 159 / 300, Logarithmic, Heikin Ashi
Quote List [2]
World Markets
Dow Jones Comp.
Settings

Problem Explanation with diagram and example

A stock broker or a person who deals with stocks can make maximum profit only when the stock is bought at minimum price and sold at maximum price. Keeping this as our basic principle we aim at solving this problem.

The test case would have the number of days the stock is active and price for each day. This problem also follows a constraint i.e., number of transactions for a given stock are limited and are given.

Say the stock prices for N days are given, and a maximum of K transactions allowed, then aim is to determine the maximum profit that can be made.

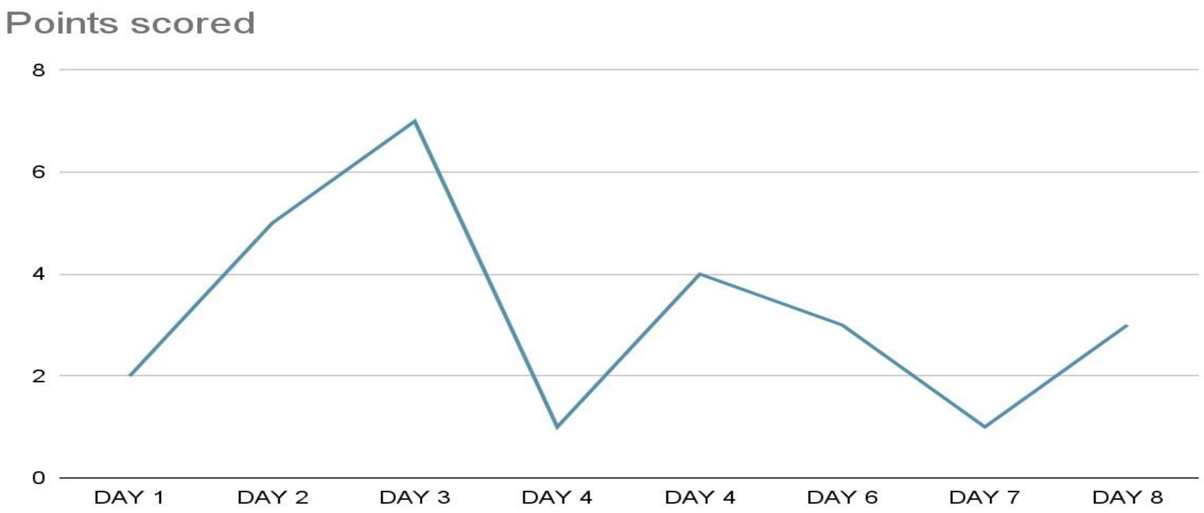
Example:

Number of days - N = 8

Maximum number of transactions - K = 3

DAY	1	2	3	4	5	6	7	8
STOCK PRICE	2	5	7	1	4	3	1	3

Explanation: The trader performs 3 transactions, the first of which is by purchasing at price 2 and selling it at price 7 followed by a purchase and sale at price 1 and 4, and by a purchase of 1 and selling at 3 respectively. Thus, the profit earned is 10. Output: 10



Design Techniques used Algorithm for the problem Explanation of algorithm with example

Approach: This approach shows how to solve this problem using Dynamic Programming Approach

ALGORITHM:

Let profit[t][i] represent maximum profit using at most t transactions up to day i (including day i). Then the relation is:

profit[t][i] = max(profit[t][i-1], max(price[i] – price[j] + profit[t-1][j]))

for all j in range [0, i-1]

profit[t][i] will be maximum of –

- profit[t][i-1] which represents not doing any transaction on the ith day.
- Maximum profit gained by selling on ith day. In order to sell shares on ith day, we need to purchase it on any one of [0, i – 1] days. If we buy shares on jth day and sell them on ith day, max profit will be price[i] – price[j] + profit[t-1][j] where j varies from 0 to i-1. Here profit[t-1][j] is best we could have done with one less transaction till jth day.

Consider the above example, using the DP approach the problem can be solved in this manner.

DAY	1	2	3	4	5	6	7	8
STOCK PRICE	2	5	7	1	4	3	1	3
	2	5	7	1	4	3	1	3
0	0	0	0	0	0	0	0	0
1	0	3	5	5	5	5	5	5
2	0	3	5	5	8	8	8	8
3	0	3	5	5	8	8	8	10

Pseudocode

```
# Python program to maximize the profit
# by doing at most k transactions
# given stock prices for n days
# Function to find out maximum profit by
# buying & selling a share at most k times
# given stock price of n days
def maxProfit(prices, n, k):

    # Bottom-up DP approach
    profit := [[0 for i in range(k + 1) do] for j in
                        range(n) do]

    # Profit is zero for the first
    # day and for zero transactions
    for i in range(1, n) do:

        for j in range(1, k + 1) do:
            max_so_far := 0

            for l in range(i) do:
                max_so_far := max(max_so_far, prices[i] prices[l]
                                  + profit[l][j - 1])
            end for loop
            profit[i][j] := max(profit[i - 1][j], max_so_far)

        End for loop

    End for loop return

    profit[n - 1][k]
```

Complexity analysis

The above solution has time complexity of O(k.n^2)

Conclusion

We use this dynamic programming approach to solve this problem and thereby optimize profits on the buying and selling of shares, this has great real-life application in the stock market but not only limited to it. We have an option to further optimize the code that is to reduce the time complexity to O(kn) by using the result from the last transaction but to keep it simple we have gone for this approach.

References

We have used help from the following sources: https://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-k-times/ https://www.techiedelight.com/find-maximum-profit-earned-at-most-k-stock-transactions/ https://www.youtube.com/watch?v=oDhu5uGq_ic