

# GeeksforGeeks

A computer science portal for geeks

## GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

## Maximum profit by buying and selling a share at most twice

In a daily share trading, a buyer buys shares in the morning and sells it on same day. If the trader is allowed to make at most 2 transactions in a day, where as second transaction can only start after first one is complete (Sell->buy->sell->buy). Given stock prices throughout day, find out maximum profit that a share trader could have made.

Examples:

Input: price[] = {10, 22, 5, 75, 65, 80}

Output: 87

Trader earns 87 as sum of 12 and 75

Buy at price 10, sell at 22, buy at 5 and sell at 80

Input: price[] = {2, 30, 15, 10, 8, 25, 80}

Output: 100  
 Trader earns 100 as sum of 28 and 72  
 Buy at price 2, sell at 30, buy at 8 and sell at 80

Input: price[] = {100, 30, 15, 10, 8, 25, 80};  
 Output: 72  
 Buy at price 8 and sell at 80.

Input: price[] = {90, 80, 70, 60, 50}  
 Output: 0  
 Not possible to earn.

**We strongly recommend to minimize your browser and try this yourself first.**

A **Simple Solution** is to consider every index 'i' and do following

Max profit with at most two transactions =  
 MAX {max profit with one transaction and subarray price[0..i] +  
 max profit with one transaction and subarray price[i+1..n-1] }  
 i varies from 0 to n-1.

Maximum possible using one transaction can be calculated using following  $O(n)$  algorithm

[Maximum difference between two elements such that larger element appears after the smaller number](#)

Time complexity of above simple solution is  $O(n^2)$ .

We can do this  $O(n)$  using following **Efficient Solution**. The idea is to store maximum possible profit of every subarray and solve the problem in following two phases.

- 1) Create a table profit[0..n-1] and initialize all values in it 0.
- 2) Traverse price[] from right to left and update profit[i] such that profit[i] stores maximum profit achievable from one transaction in subarray price[i..n-1]
- 3) Traverse price[] from left to right and update profit[i] such that profit[i] stores maximum profit such that profit[i] contains maximum achievable profit from two transactions in subarray price[0..i].
- 4) Return profit[n-1]

To do step 1, we need to keep track of maximum price from right to left side and to do step 2, we need to keep track of minimum price from left to right. Why we traverse in reverse directions? The idea is to save space, in second step, we use same array for both purposes, maximum with 1 transaction and maximum with 2 transactions. After an iteration i, the array profit[0..i] contains maximum profit with 2 transactions and profit[i+1..n-1] contains profit with two transactions.

Below is C++ implementation of above idea.

```
// C++ program to find maximum possible profit with at most
// two transactions
#include<iostream>
using namespace std;

// Returns maximum profit with two transactions on a given
// list of stock prices, price[0..n-1]
```

```

int maxProfit(int price[], int n)
{
    // Create profit array and initialize it as 0
    int *profit = new int[n];
    for (int i=0; i<n; i++)
        profit[i] = 0;

    /* Get the maximum profit with only one transaction
       allowed. After this loop, profit[i] contains maximum
       profit from price[i..n-1] using at most one trans. */
    int max_price = price[n-1];
    for (int i=n-2; i>=0; i--)
    {
        // max_price has maximum of price[i..n-1]
        if (price[i] > max_price)
            max_price = price[i];

        // we can get profit[i] by taking maximum of:
        // a) previous maximum, i.e., profit[i+1]
        // b) profit by buying at price[i] and selling at
        //     max_price
        profit[i] = max(profit[i+1], max_price-price[i]);
    }

    /* Get the maximum profit with two transactions allowed
       After this loop, profit[n-1] contains the result */
    int min_price = price[0];
    for (int i=1; i<n; i++)
    {
        // min_price is minimum price in price[0..i]
        if (price[i] < min_price)
            min_price = price[i];

        // Maximum profit is maximum of:
        // a) previous maximum, i.e., profit[i-1]
        // b) (Buy, Sell) at (min_price, price[i]) and add
        //     profit of other trans. stored in profit[i]
        profit[i] = max(profit[i-1], profit[i] +
                        (price[i]-min_price) );
    }
    int result = profit[n-1];

    delete [] profit; // To avoid memory leak

    return result;
}

// Drive program
int main()
{
    int price[] = {2, 30, 15, 10, 8, 25, 80};
    int n = sizeof(price)/sizeof(price[0]);
    cout << "Maximum Profit = " << maxProfit(price, n);
}

```

```

    return 0;
}

```

Output:

Maximum Profit = 100

Time complexity of the above solution is  $O(n)$ .

Algorithmic Paradigm: Dynamic Programming

This article is contributed by **Amit Jaiswal**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

## Related Topics:

- [Find Union and Intersection of two unsorted arrays](#)
- [Pythagorean Triplet in an array](#)
- [Design a data structure that supports insert, delete, search and getRandom in constant time](#)
- [Print missing elements that lie in range 0 – 99](#)
- [Iterative Merge Sort](#)
- [Group multiple occurrence of array elements ordered by first occurrence](#)
- [Given a sorted and rotated array, find if there is a pair with a given sum](#)
- [Divide and Conquer | Set 7 \(The Skyline Problem\)](#)

Tags: [Dynamic Programming](#)



Tweet

4

+1

1

Writing code in comment? Please use [ideone.com](http://ideone.com) and share the link here.

20 Comments

GeeksforGeeks

1 Login ▾

♥ Recommend

🔗 Share

Sort by Newest ▾



Join the discussion...



**Raj** • 4 days ago

//Without DP.

```
#include<iostream>
```

```
using namespace std;
```

```
int BuySell(int arr[],int n){
```

```
int first=0;
```

```
int second=0;
```

```
int sum=0;
```

```
for(int i=0;i<n;i++){ int="" j="i;" while(i<n="" &&="" arr[i]<arr[i+1]){="" i++;" }="" if(i!="j){"
```

```

sum="arr[i]-arr[j];" }="" cout<<"sum=" is="" "<<sum<<endl;="" if(sum="">first)
first=sum;
else if(sum>second)
second=sum;

}
//cout<<"Value is "<<first<<":"<<second; return="" first+second;="" }="" drive="" program=""
int="" main()="" {="" int="" price[]={2," 30,="" 15,="" 10,="" 8,="" 25,="" 80};="" int=""
n="sizeof(price)/sizeof(price[0]);" cout<<"max="" profit="" is="" "<<buysell(price,="" n)
<<endl;="" return="" 0;="" }="">

```

^ | v • Reply • Share ›

**GeeksforGeeks** Mod ➔ Raj • 4 days ago

Raj, could you please post the code on [ideone.com](http://ideone.com) and share the link here.

^ | v • Reply • Share ›

**Rahul** ➔ GeeksforGeeks • a day ago

Hey @GeeksForGeeks. Isnt this solution also applicable to this problems

<http://www.geeksforgeeks.org/s...>

^ | v • Reply • Share ›

**Saras\_Arya** • 8 days ago

How will you extend it for at most 3 transactions or lets say n transactions. What would be the approach?

^ | v • Reply • Share ›

**pruthvi raj** • 17 days ago

I think this will work for exactly 2 transactions. Will this also work for more than 2 transactions?

^ | v • Reply • Share ›

**amit jaiswal** ➔ pruthvi raj • 16 days ago

Yes

This is for at most two transactions.

^ | v • Reply • Share ›



**Lakshman** ➔ amit jaiswal • 13 days ago

Can we sell and buy at the same price for the second transaction? the code seems to allow that case. Please let me know if that is what is intended

^ | v • Reply • Share ›

**Abhijeet Sachdev** ➔ Lakshman • 3 days ago

Yes you are right. . The code is bit confusing actually. . from the code it looks like that the same price is being used again. . but it will never happen : Reason: because the same price "can never" be at the farthest

distance from maximum in RHS and minimum in LHS. . kyuki aga aisa hota tow. . jo minimum hai LHS me wo khud he RHS ke maximum se farthest ho jata

^ | v • Reply • Share ›



**Lakshman** → Lakshman • 13 days ago

Meaning .. sell first transaction and buy for the second transaction at the same price.. the code seems to allow that .

^ | v • Reply • Share ›

**Manish** • 18 days ago

If you compare the problem with biggest next integer in an array, then a simple  $O(n)$  solution exists, with max  $O(1)$  as storage complexity.

^ | v • Reply • Share ›

**pruthvi raj** • 18 days ago

For input 3,4,7,10, output should be 4 but the efficient solution gives 7 as output. Please correct me if am wrong.

^ | v • Reply • Share ›



**weiming** → pruthvi raj • 18 days ago

“the trader is allowed to make at most 2 transactions in a day”

It's "at most", not "exactly". The solution is buying at price 3 and selling at 10, so profit = 7.

^ | v • Reply • Share ›

**pruthvi raj** → weiming • 17 days ago

yeahh... thanx

^ | v • Reply • Share ›

**palindname** • 18 days ago

Input size should be EVEN to my understanding of the question. Sample and driver program has incorrect input in such case.

Since the input is an alternate BUY/SELL cost, i doubt the way  $\text{profit}[i]$  is evaluated in Step 2. Tried executing the code with

```
int price[] = {2, 30, 15, 20, 8, 10, 90, 80};
```

Output : 110

I think it should be 100 though. Please correct me if I am wrong.

^ | v • Reply • Share ›

**GeeksforGeeks** Mod → palindname • 18 days ago

110 seems to be correct answe:

Buy at price 2, sell at 30, buy at 8 and sell at 90. Profit =  $(30 - 2) + (90 - 8) = 28 + 82 = 110$

^ | v • Reply • Share ›

**palindname** → GeeksforGeeks • 17 days ago

Buy->Sell->Buy->Sell

Shouldn't 90 be a buying price? [My first concern]

^ | v • Reply • Share ›

**Uponder Reddy** → GeeksforGeeks • 17 days ago

is 90 selling price

^ | v • Reply • Share ›



**weiming** • 18 days ago

generic problem

<https://leetcode.com/problems/...>

^ | v • Reply • Share ›

**Deepesh Maheshwari** → weiming • 16 days ago

Amit Jaiswal please explain algo approach for this generic case.

^ | v • Reply • Share ›

**amit jaiswal** → weiming • 16 days ago

Can this be done by creating  $N \times K$  table where  $table[i][j]$  is max profit till profit[i].

And then find max from the matrix?

^ | v • Reply • Share ›

Subscribe

Add Disqus to your site

Privacy

- 
- 
- 
- 
- - [Interview Experiences](#)
  - [Advanced Data Structures](#)
  - [Dynamic Programming](#)
  - [Greedy Algorithms](#)
  - [Backtracking](#)
  - [Pattern Searching](#)
  - [Divide & Conquer](#)
  - [Mathematical Algorithms](#)
  - [Recursion](#)
  - [Geometric Algorithms](#)
- 

## • Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks

## • Recent Comments

- It\_k  
i need help for coding this function in java...  
[Java Programming Language](#) · [2 hours ago](#)
- Piyush  
What is the purpose of else if (recStack[\*i])...  
[Detect Cycle in a Directed Graph](#) · [2 hours ago](#)
- Andy Toh  
My compile-time solution, which agrees with the...



[Dynamic Programming | Set 16 \(Floyd Warshall Algorithm\)](#) · [2 hours ago](#)

- [lucy](#)

because we first fill zero in first col and...

[Dynamic Programming | Set 29 \(Longest Common Substring\)](#) · [2 hours ago](#)

- [lucy](#)

@GeeksforGeeks i don't n know what is this long...

[Dynamic Programming | Set 28 \(Minimum insertions to form a palindrome\)](#) · [3 hours ago](#)

- [manish](#)

Because TAN is not a subsequence of RANT. ANT...

[Given two strings, find if first string is a subsequence of second](#) · [3 hours ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team