



Practice &gt; Algorithms &gt; Dynamic Programming &gt; Stock Maximize

# Stock Maximize ☆

**Problem**

## Submissions

## Leaderboard

## Discussions

## Editorial

## Topics

Your algorithms have become so good at predicting the market that you now know what the share price of Wooden Orange Toothpicks Inc. (WOT) will be for the next number of days.

Each day, you can either buy one share of WOT, sell any number of shares of WOT that you own, or not make any transaction at all. What is the maximum profit you can obtain with an optimum trading strategy?

For example, if you know that prices for the next two days are *prices* = [1, 2], you should buy one share day one, and sell it day two for a profit of 1. If they are instead *prices* = [2, 1], no profit can be made so you don't buy or sell stock those days.

**Input Format**

The first line contains the number of test cases *t*.

Each of the next *t* pairs of lines contain:

- The first line contains an integer *n*, the number of predicted prices for WOT.
- The next line contains *n* space-separated integers *prices*[*i*], each a predicted stock price for day *i*.

**Constraints**

- $1 \leq t \leq 10$
- $1 \leq n \leq 50000$
- $1 \leq \text{prices}[i] \leq 100000$

**Output Format**

Output *t* lines, each containing the maximum profit which can be obtained for the corresponding test case.

**Sample Input**

```
3
3
5 3 2
3
1 2 100
4
1 3 1 2
```

**Sample Output**

```
0
197
3
```

**Explanation**

For the first case, you cannot obtain any profit because the share price never rises.

For the second case, you can buy one share on the first two days and sell both of them on the third day.

Author [HackerRank](#)Difficulty [Medium](#)

Max Score 50

Submitted By [23399](#)

## NEED HELP?

[View discussions](#) [View editorial](#) [View top submissions](#)

## RESOURCES

[Dynamic Programming Basics](#)



## RATE THIS CHALLENGE






## MORE DETAILS

[Download problem statement](#) [Download sample test cases](#) [Suggest Edits](#)

For the third case, you can buy one share on day 1, sell one on day 2, buy one share on day 3, and sell one share on day 4.


**Current Buffer** (saved locally, editable)  

Java 7 

```
1 ▼ import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 ▼ public class Solution {
8
9 ▼     static int stockmax(int[] prices) {
10         // Complete this function
11     }
12
13 ▼     public static void main(String[] args) {
14         Scanner in = new Scanner(System.in);
15         int t = in.nextInt();
16 ▼         for(int a0 = 0; a0 < t; a0++){
17             int n = in.nextInt();
18 ▼             int[] prices = new int[n];
19 ▼             for(int prices_i = 0; prices_i < n; prices_i++){
20 ▼                 prices[prices_i] = in.nextInt();
21             }
22             int result = stockmax(prices);
23             System.out.println(result);
24         }
25         in.close();
26     }
27 }
```

Line: 1 Col: 1

 Upload Code as File ☐ Test against custom input 

Run Code

Submit Code