



Experiment 3.3

Student Name: Harshit Raj

UID: 20BCS9266

Branch: CSE

Section/Group: 608/A

Semester: 6

Date of Performance: 03/05/2023

Subject Name: Competitive Coding-II

Subject Code: 20CSP-351

Aim: To demonstrate the concept of Dynamic Programming

Objective:

Best time to buy and sell the stock: <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

121. Best Time to Buy and Sell Stock

Easy 25K 786

Companies

You are given an array `prices` where `prices[i]` is the price of a given stock on the i^{th} day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock.

Return the *maximum profit* you can achieve from this transaction. If you cannot achieve any profit, return `0`.

Example 1:

```
Input: prices = [7,1,5,3,6,4]
Output: 5
Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.
Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.
```

Example 2:

```
Input: prices = [7,6,4,3,1]
Output: 0
Explanation: In this case, no transactions are done and the max profit = 0.
```

Code:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
class Solution {           public int
maxProfit(int[] prices) {   int lsf
= Integer.MAX_VALUE;      int op = 0;
int pist = 0;

    for(int i = 0; i < prices.length; i++){
if(prices[i] < lsf){        lsf =
prices[i];
    }
    pist = prices[i] - lsf;
if(op < pist){              op =
pist;
    }
    }           return
op;
    }
}
```

Java

Runtime 2 ms

Beats 92.79%

Memory 59.6 MB

Beats 20.34%

[Click the distribution chart to view more details](#)

Notes

Write your notes here

Related Tags

Select tags

0/5

```
class Solution {
    public int maxProfit(int[] prices) {
        int lsf = Integer.MAX_VALUE;
        int op = 0;
        int pist = 0;

        for(int i = 0; i < prices.length; i++){
            if(prices[i] < lsf){
                lsf = prices[i];
            }
            pist = prices[i] - lsf;
            if(op < pist){
                op = pist;
            }
        }
        return op;
    }
}
```

Console ^

XX

Run

Submit

Climbing Stairs: <https://leetcode.com/problems/climbing-stairs/>

70. Climbing Stairs

Hint 

Easy

17.9K

562



 Companies

You are climbing a staircase. It takes `n` steps to reach the top.

Each time you can either climb `1` or `2` steps. In how many distinct ways can you climb to the top?

Example 1:

Input: `n = 2`

Output: `2`

Explanation: There are two ways to climb to the top.

1. 1 step + 1 step
2. 2 steps

Example 2:

Input: `n = 3`

Output: `3`

Explanation: There are three ways to climb to the top.

1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step

Code:

```
class Solution { public:
    int climbStairs(int n) {
        if (n <= 2) return n;
        int prev = 2, prev2 = 1, res;
        for (int i = 3; i <= n; i++) {
            res = prev + prev2;
            prev2 = prev;
            prev = res;
        }
        return res;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

C++



Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags

0/5

```
class Solution {
public:
    int climbStairs(int n) {
        if (n <= 2) return n;
        int prev = 2, prev2 = 1, res;
        for (int i = 3; i <= n; i++) {
            res = prev + prev2;
            prev2 = prev;
            prev = res;
        }
        return res;
    }
};
```

Console ^



Run

Submit