|  |  |
| --- | --- |
| **Experiment-3.1** | |
| **Student Name: Shubham Kumar** | **UID: 21BCS7104** |
| **Branch: CSE** | **Section/Group: 614-A** |
| **Semester: 6th** | **Date of Performance: 27-03-2024** |
| **Subject Name: Advanced Programming Lab-2** | **Subject Code: 21CSP-351** |

**1. Aim:** Greedy Algorithm.

A) Return the minimum number of candies you need to have to distribute the candies to the children.(135. Candy)

B) Find and return the maximum profit you can achieve.( 122. Best Time to Buy and Sell Stock II)

**2. Source Code/Output:**

**A) Code:**

class Solution {

public:

int candy(vector<int>& ratings) {

const int n = ratings.size();

int ans = 0;

vector<int> l(n, 1);

vector<int> r(n, 1);

for (int i = 1; i < n; ++i)

if (ratings[i] > ratings[i - 1])

l[i] = l[i - 1] + 1;

for (int i = n - 2; i >= 0; --i)

if (ratings[i] > ratings[i + 1])

r[i] = r[i + 1] + 1;

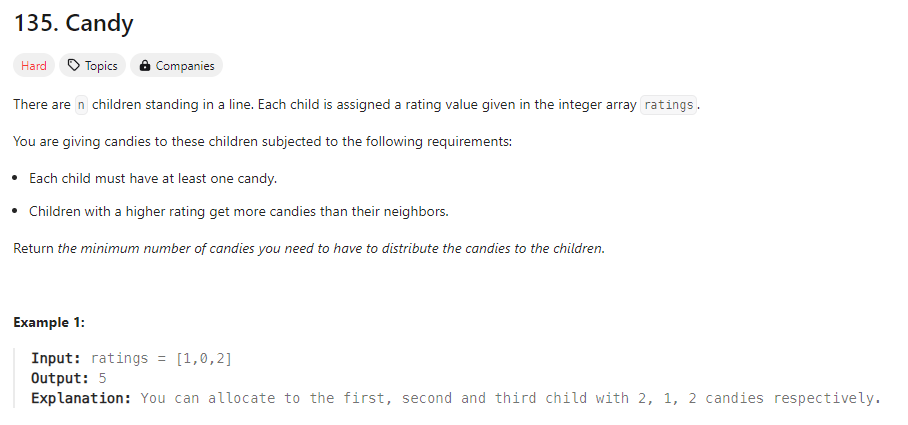
for (int i = 0; i < n; ++i)

ans += max(l[i], r[i]);

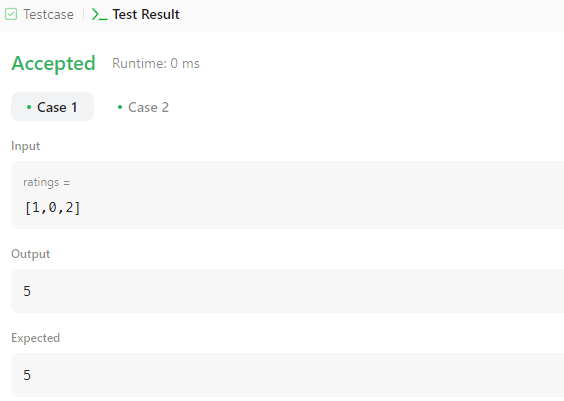
return ans;

}};

**Description:**

****

**Output:**



**B)** **Code:**

class Solution {

public:

int maxProfit(vector<int>& prices) {

int sell = 0;

int hold = INT\_MIN;

for (const int price : prices) {

sell = max(sell, hold + price);

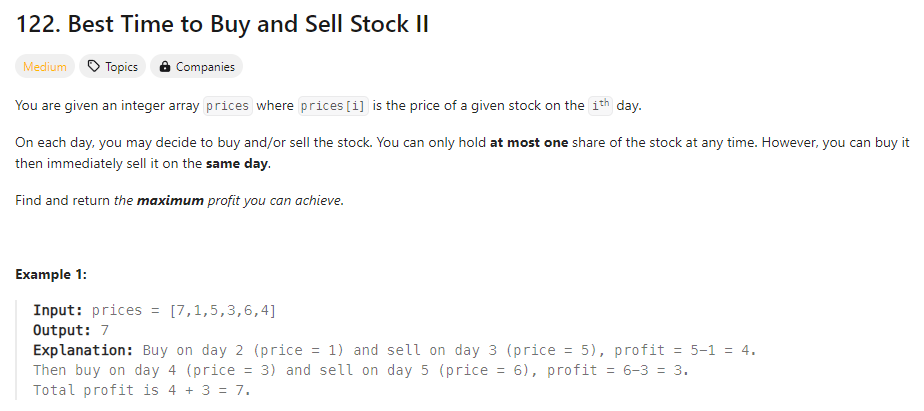
hold = max(hold, sell - price);

}

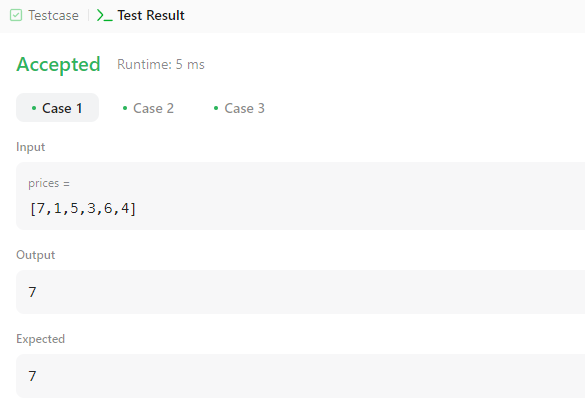
return sell;

}};

**Description:**



**Output:**

****