



EXPERIMENT - 8

Student Name: Sameer

UID: 22BCS15631

Branch: Computer Science & Engineering

Section/Group: IOT-614/B

Semester: 6th

Date of Performance: 10/04/2025

Subject Name: Advanced Programming Lab-2

Subject Code: 22CSP-351

Q.1. Minimum Operations to Make the Array Increasing

You are given an integer array `nums` (0-indexed). In one operation, you can choose an element of the array and increment it by 1.

For example, if `nums = [1,2,3]`, you can choose to increment `nums[1]` to make `nums = [1,3,3]`.

Return the minimum number of operations needed to make `nums` strictly increasing.

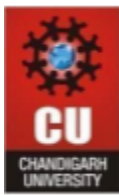
An array `nums` is strictly increasing if `nums[i] < nums[i+1]` for all $0 \leq i < \text{nums.length} - 1$. An array of length 1 is trivially strictly increasing.

Code:

```
class Solution {
public:
    int minOperations(vector<int>& nums)
    {
        int op = 0 ;

        for (int i = 1 ; i < nums.size() ; i++)
        {
            if (nums[i] <= nums[i - 1])
            {
                int inc = nums[i - 1] - nums[i] + 1 ;
                nums[i] = nums[i] + inc ;
                op = op + inc ;
            }
        }

        return op ;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

☒ Testcase | [Test Result](#)

Accepted Runtime: 0 ms

- Case 1
- Case 2
- Case 3

Input
nums =
[1,1,1]

Output
3

Expected
3

Accepted 94 / 94 testcases passed

Sameer submitted at Apr 10, 2025 15:26

[Solution](#)

Runtime

15 ms | Beats 20.13%

[Analyze Complexity](#)

Memory

19.49 MB | Beats 83.86%



Q.2. Maximum Score From Removing Substrings

You are given a string s and two integers x and y . You can perform two types of operations any number of times.

- Remove substring "ab" and gain x points.

For example, when removing "ab" from "cabxbae" it becomes "cxbae".

- Remove substring "ba" and gain y points.

For example, when removing "ba" from "cabxbae" it becomes "cabxe".

Return the maximum points you can gain after applying the above operations on s .

Code:

```
class Solution {
public:
    int maximumGain(string s, int x, int y) {
        if (x > y)
            return process(s, 'a', 'b', x, y);
        else
            return process(s, 'b', 'a', y, x);
    }

private:
    int process(string s, char first, char second, int firstVal,
               int secondVal) {
        int score = 0;
        stack<char> st;
        string temp;

        for (char ch : s) {
            if (!st.empty() && st.top() == first && ch == second) {
                st.pop();
                score += firstVal;
            } else {
                st.push(ch);
            }
        }

        while (!st.empty()) {
            temp += st.top();
            st.pop();
        }
        reverse(temp.begin(), temp.end());
        return score;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        st.pop();
    }
    reverse(temp.begin(), temp.end());

    for (char ch : temp) {
        if (!st.empty() && st.top() == second && ch == first) {
            st.pop();
            score += secondVal;
        } else {
            st.push(ch);
        }
    }

    return score;
}
};
```

Output:

Accepted Runtime: 0 ms

• Case 1

• Case 2

Input

s =
"cdbcbbaaabab"

x =
4

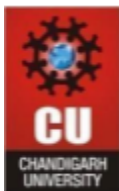
y =
5

Output

19

Expected

19



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Accepted 77 / 77 testcases passed

Sameer submitted at Apr 10, 2025 15:39

Editorial

Solution

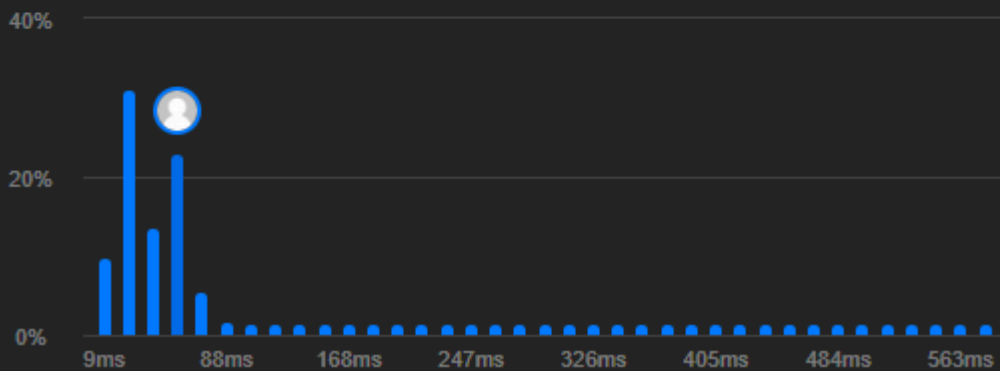
Runtime

63 ms | Beats 27.92%

Analyze Complexity

Memory

31.30 MB | Beats 17.29%





Q.3. Minimum Operations to Make a Subsequence

You are given an array `target` that consists of distinct integers and another integer array `arr` that can have duplicates.

In one operation, you can insert any integer at any position in `arr`. For example, if `arr = [1,4,1,2]`, you can add 3 in the middle and make it `[1,4,3,1,2]`. Note that you can insert the integer at the very beginning or end of the array.

Return the minimum number of operations needed to make `target` a subsequence of `arr`.

A subsequence of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the remaining elements' relative order. For example, `[2,7,4]` is a subsequence of `[4,2,3,7,2,1,4]` (the underlined elements), while `[2,4,2]` is not.

Code:

```
class Solution {
public:
    int minOperations(vector<int>& target, vector<int>& arr) {
        unordered_map<int, int> pos;

        for (int i = 0; i < target.size(); ++i)
            pos[target[i]] = i;

        vector<int> lis;
        for (int num : arr) {
            if (pos.find(num) != pos.end()) {
                int idx = pos[num];

                auto it = lower_bound(lis.begin(), lis.end(), idx);

                if (it == lis.end()) {
                    lis.push_back(idx);
                }

                else {
                    *it = idx;
                }
            }
        }

        return target.size() - lis.size();
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
}  
};
```

Output:

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

target =
[5,1,3]

arr =
[9,4,2,3,4]

Output

2

Expected

2

Accepted 82 / 82 testcases passed

Sameer submitted at Apr 10, 2025 15:50

[Solution](#)

Runtime

102 ms | Beats **96.49%** 🌿

🔮 [Analyze Complexity](#)

Memory

113.84 MB | Beats **75.88%** 🌿



Q.4. Maximum Number of Tasks You Can Assign

You have n tasks and m workers. Each task has a strength requirement stored in a 0-indexed integer array `tasks`, with the i th task requiring `tasks[i]` strength to complete. The strength of each worker is stored in a 0-indexed integer array `workers`, with the j th worker having `workers[j]` strength. Each worker can only be assigned to a single task and must have a strength greater than or equal to the task's strength requirement (i.e., `workers[j] >= tasks[i]`).

Additionally, you have pills magical pills that will increase a worker's strength by `strength`. You can decide which workers receive the magical pills, however, you may only give each worker at most one magical pill.

Given the 0-indexed integer arrays `tasks` and `workers` and the integers `pills` and `strength`, return the maximum number of tasks that can be completed.

Code:

```
class Solution {
public:
    bool canAssign(int k, vector<int>& tasks, vector<int>& workers, int pills,
                  int strength) {
        multiset<int> available(workers.end() - k, workers.end());

        int usedPills = 0;

        for (int i = k - 1; i >= 0; --i) {
            int t = tasks[i];

            auto it = available.lower_bound(t);
            if (it != available.end()) {
                available.erase(it);
            }

            else {
                if (usedPills == pills)
                    return false;

                it = available.lower_bound(t - strength);
                if (it == available.end())
                    return false;

                available.erase(it);
            }
        }
    }
};
```




```
        usedPills++;
    }
}

return true;
}

int maxTaskAssign(vector<int>& tasks, vector<int>& workers, int pills,
                 int strength) {
    sort(tasks.begin(), tasks.end());
    sort(workers.begin(), workers.end());

    int left = 0, right = min((int)tasks.size(), (int)workers.size()),
        answer = 0;

    while (left <= right) {
        int mid = left + (right - left) / 2;

        if (canAssign(mid, tasks, workers, pills, strength)) {
            answer = mid;
            left = mid + 1;
        }

        else {
            right = mid - 1;
        }
    }

    return answer;
}
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output:

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

tasks =
[3, 2, 1]

workers =
[0, 3, 3]

pills =
1

strength =
1

Output

3

Accepted 49 / 49 testcases passed

Sameer submitted at Apr 10, 2025 15:55

Solution

Runtime

745 ms | Beats **54.94%**

[Analyze Complexity](#)

Memory

286.37 MB | Beats **56.91%**