ASSIGNMENT - 4

Student Name: Manreet Kaur **UID: 22BCS15550**

Branch: BE-CSE **Section/Group: 608/B**

Semester: 6th **Subject Name: AP LAB**

1. Longest Nice Substring

Problem List

Description | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
3	Accepted Mar 06, 2025	C++	361 ms	122.6 MB	
2	Accepted Feb 28, 2025	C++	4 ms	11.5 MB	
1	Accepted Feb 27, 2025	C++	366 ms	122.6 MB	

Code

C++ Auto

```cpp
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n=s.length();
        if (s.length()<2) {
            return "";
        }
        bool lower[26]={false};
        bool upper[26]={false};
        for(char c:s){
            if(islower(c)){
                lower[c-'a']=true;
            }
            else{
                upper[c-'A']=true;
            }
        }
        for(int i=0;i<n;i++){
            char c=s[i];
            if(islower(c)&&!upper[c-'a']){
            string left=longestNiceSubstring(s.substr(0,i));
            string right=longestNiceSubstring(s.substr(i+1));
            return left.length()>=right.length()?left:right;
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions

Progress | Points

Try New Features

Orders

My Playgrounds

Settings

Classic Mode

2. Reverse Bits

Problem List

Description | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
1	Accepted 31 minutes ago	C++	0 ms	7.7 MB	

Code

C++ Auto

```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
    uint32_t result = 0;
    for (int i = 0; i < 32; ++i) {
        result = (result << 1) | (n & 1);
        n >>= 1;
    }
    return result;
}
};
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions | Progress | Points

3. Number of 1 Bits

Problem List

Description | Accepted | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
2	Accepted 27 minutes ago	C++	0 ms	8.2 MB	
1	Accepted 30 minutes ago	C++	0 ms	8.2 MB	

Code

C++ Auto

```cpp
class Solution {
public:
    int hammingWeight(uint32_t n) {
    int count = 0;
    while (n) {
        n &= (n - 1);
        count++;
    }
    return count;
}
};
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions | Progress | Points

4. Maximum Subarray

Problem List

Description | Accepted | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
1	Accepted 26 minutes ago	C++	0 ms	71.8 MB	

Code

C++ Auto

```cpp
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
    int max_sum = nums[0];
    int current_sum = 0;

    for (int num : nums) {
        current_sum += num;
        max_sum = max(max_sum, current_sum);
        if (current_sum < 0) current_sum = 0;
    }

    return max_sum;
}
};
```

Manreet_05

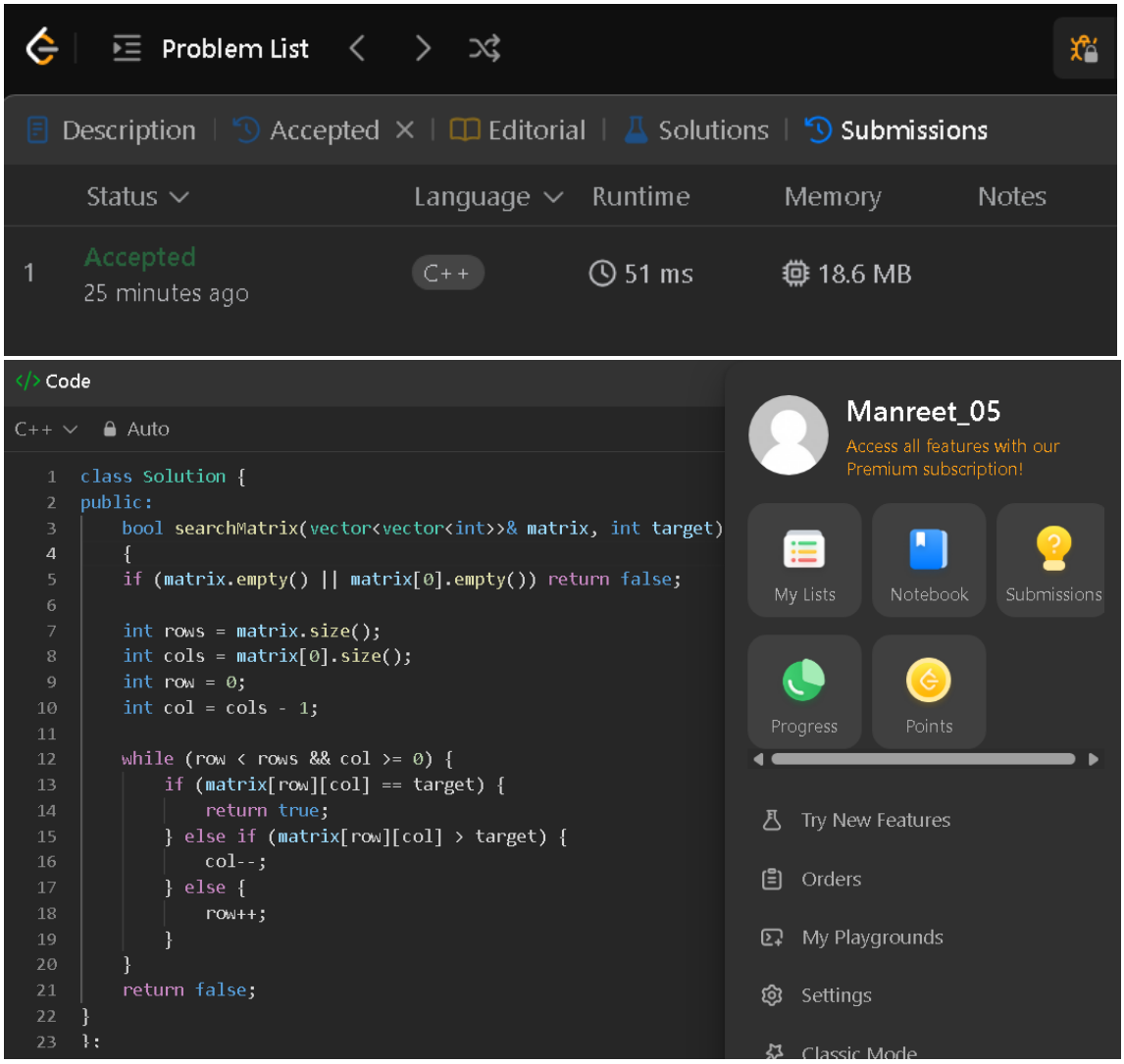
Access all features with our Premium subscription!

My Lists | Notebook | Submissions | Progress | Points

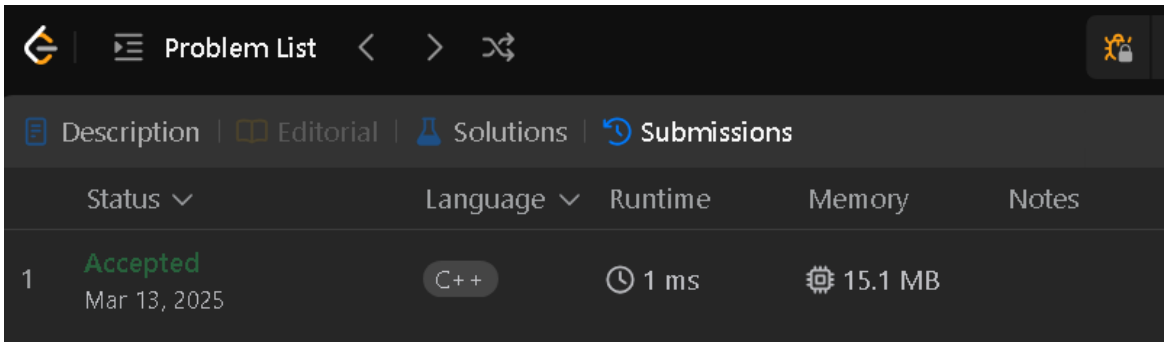
Try New Features

Orders

5. Search a 2D Matrix II



6. Super Pow



Code | Accepted

C++ Auto

```cpp
class Solution {
public:
const int MOD = 1337;
int powerMod(int a, int k) {
    a %= MOD;
    int result = 1;
    while (k > 0) {
        if (k % 2 == 1) {
            result = (result * a) % MOD;
        }
        a = (a * a) % MOD;
        k /= 2;
    }
    return result;
}
int superPow(int a, vector<int>& b) {
    if (b.empty()) return 1;

    int lastDigit = b.back();
    b.pop_back();

    int part1 = powerMod(a, lastDigit);
    int part2 = powerMod(superPow(a, b), 10);
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions

Progress | Points

Try New Features

Orders

My Playgrounds

Settings

Classic Mode

7. Beautiful Array

Code

C++ Auto

```cpp
class Solution {
public:
    vector<int> beautifulArray(int n) {
    vector<int> result = {1};
    while (result.size() < n) {
        vector<int> temp;
        for (int x : result) {
            if (2 * x - 1 <= n) {
                temp.push_back(2 * x - 1);
            }
        }
        for (int x : result) {
            if (2 * x <= n) {
                temp.push_back(2 * x);
            }
        }
        result = temp;
    }
    return result;
    }
};
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions

Progress | Points

Try New Features

Orders

My Playgrounds

Settings

Problem List

Description | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
1	Accepted 12 minutes ago	C++	0 ms	10.1 MB	

8. The Skyline Problem

Problem List

Description | Accepted | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
1	Accepted 20 minutes ago	C++	159 ms	105.3 MB	

Code

C++ Auto

```cpp
class Solution {
public:
    vector<vector<int>> mergeSkylines(vector<vectc
    vector<vector<int>> merged;
    int h1 = 0, h2 = 0;
    int i = 0, j = 0;

    while (i < left.size() && j < right.size()) {
        int x, h;
        if (left[i][0] < right[j][0]) {
            x = left[i][0];
            h1 = left[i][1];
            h = max(h1, h2);
            i++;
        } else if (left[i][0] > right[j][0]) {
            x = right[j][0];
            h2 = right[j][1];
            h = max(h1, h2);
            j++;
        } else {
            x = left[i][0];
            h1 = left[i][1];
            h2 = right[j][1];
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions

Progress | Points

Try New Features

Orders

My Playgrounds

Settings

Classic Mode

9. Reverse Pairs

Problem List

Description | Accepted | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
1	Accepted 9 minutes ago	C++	545 ms	243.5 MB	

Code

C++ Auto

```cpp
class Solution {
public:

int mergeAndCount(vector<int>& nums, int left, int mi
    int count = 0;
    int j = mid + 1;

    for (int i = left; i <= mid; i++) {
        while (j <= right && nums[i] > 2LL * nums[j])
            j++;
        }
        count += (j - (mid + 1));
    }

    vector<int> temp;
    int i = left, k = mid + 1;

    while (i <= mid && k <= right) {
        if (nums[i] <= nums[k]) {
            temp.push_back(nums[i++]);
        } else {
            temp.push_back(nums[k++]);
        }
```

Manreet_05

Access all features with our Premium subscription!

My Lists | Notebook | Submissions | Progress | Points

Try New Features

Orders

My Playgrounds

Settings

10. Longest Increasing Subsequence II

Problem List

Description | Editorial | Solutions | Submissions

	Status	Language	Runtime	Memory	Notes
2	Accepted 5 minutes ago	C++	26 ms	54.5 MB	

Code

C++ Auto

```cpp
class SegmentTree {
public:
    vector<int> tree;
    int size;

    SegmentTree(int n) {
        size = n;
        tree.resize(2 * n, 0);
    }

    void update(int index, int value) {
        index += size;
        tree[index] = max(tree[index], value);
        while (index > 1) {
            index /= 2;
            tree[index] = max(tree[2 * index], tree[2 * index + 1]);
        }
    }
    int query(int left, int right) {
        int result = 0;
        left += size;
        right += size;
        while (left < right) {
```