

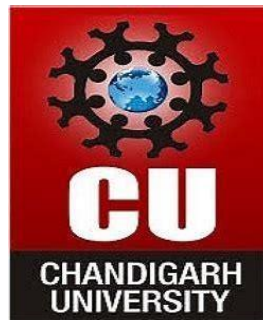


**CHANDIGARH
UNIVERSITY**
Discover. Learn. Empower.

**NAAC
GRADE A+**
Accredited University

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering
(BE-CSE-6th Sem)



Subject Name: Advanced Programming Lab - 2

Subject Code: 22CSP-351

Submitted to: Vishal Sir

Submitted by:

Name: Jaideep Singh

UID: 22BCS16802

Section: FL-IOT-604

Group: A

1763. Longest Nice Substring

Problem List < > 🔍

Run Submit ⌛

0 0 0 Premium

Description | Editorial | Solutions | Submissions | Accepted ✕

< All Submissions 🔗

Accepted 73 / 73 testcases passed
Jaideep Singh submitted at Feb 05, 2025 15:37 [Solution](#)

Runtime ⌚
0 ms | Beats 100.00% 🏆
[Analyze Complexity](#)

Memory 📦
8.27 MB | Beats 96.75% 🏆

Runtime (ms)	Percentage (%)
2ms	~35%

Code | C++

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        string output = "";
        int count = 0;
        for(int i = 0; i < s.length(); i++){
            int smallMask = 0;
            int largeMask = 0;
            char ch = s[i];
            int chint = 0;
            if(ch >= 65 && ch <= 90){
                chint = ch - 'A';
            }
        }
    }
};
```

</> Code

```
1 class Solution {
2 public:
3     string longestNiceSubstring(string s) {
4         string output = "";
5         int count = 0;
6         for(int i = 0; i < s.length(); i++){
7             int smallMask = 0;
8             int largeMask = 0;
9             char ch = s[i];
10            int chint = 0;
11            if(ch >= 65 && ch <= 90){
12                chint = ch - 'A';
13            }
14        }
15    }
16};
```

Testcase > Test Result

Case 1 Case 2 Case 3 +

s =

"YazaAay"

</> Source ⓘ

190. Reverse Bits

Description | Editorial | Accepted ✕ | Solutions | Submissions

< > 🔍

Run Submit ⌛

0 0 0 Premium

< All Submissions 🔗

Accepted 600 / 600 testcases passed
Jaideep Singh submitted at Feb 05, 2025 15:40 [Editorial](#) [Solution](#)

Runtime ⌚
4 ms | Beats 33.11%
[Analyze Complexity](#)

Memory 📦
7.88 MB | Beats 29.58%

Runtime (ms)	Percentage (%)
4ms	~40%

Code | C++

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t ans = 0;
        int i = 32;
        while(i--){
            ans <<= 1;
            ans |= n & 1;
            n >>= 1;
        }
        return ans;
    }
};
```

</> Code

```
1 class Solution {
2 public:
3     uint32_t reverseBits(uint32_t n) {
4         uint32_t ans = 0;
5         int i = 32;
6         while(i--){
7             ans <<= 1;
8             ans |= n & 1;
9             n >>= 1;
10        }
11        return ans;
12    }
13};
```

Testcase > Test Result

Case 1 Case 2 +

n =

00000010100101000001111010011100

</> Source ⓘ

191. Number of 1 Bits

Problem List < > 🔍

Run Submit ⌛

0 0 Premium

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 598 / 598 testcases passed

Jaideep Singh submitted at Feb 05, 2025 15:41

Editorial Solution

Runtime 0 ms | Beats 100.00%

Memory 8.27 MB | Beats 47.62%

Analyze Complexity

Runtime (ms)	Beats (%)
0	100.00%
1	~0%
2	~0%
3	~0%
4	~0%

Code | C++

```
class Solution {
public:
    int hammingWeight(int n) {
        int c=0,i=0,m=n;
        long bit;
        while(m>0)
        {
            bit=pow(2,i);
            if((n&bit)>=1)
                c++;
            m/=2;
            i++;
        }
    }
};
```

Code

```
1 class Solution {
2 public:
3     int hammingWeight(int n) {
4         int c=0,i=0,m=n;
5         long bit;
6         while(m>0)
7         {
8             bit=pow(2,i);
9             if((n&bit)>=1)
10                 c++;
11             m/=2;
12             i++;
13         }
14     }
15 }
```

Testcase Test Result

Case 1 Case 2 Case 3 +

n =

11

</> Source

53. Maximum Subarray

Problem List < > 🔍

Run Submit ⌛

0 0 Premium

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 210 / 210 testcases passed

Jaideep Singh submitted at Sep 04, 2024 18:41

Editorial Solution

Runtime 85 ms | Beats 5.49%

Memory 70.40 MB | Beats 99.93%

Analyze Complexity

Runtime (ms)	Beats (%)
85	5.49%
1	~0%
2	~0%
3	~0%
4	~0%
5	~0%

Code | C++

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxsum = INT_MIN;
        int sum=0;
        int size= nums.size();
        for (int i=0; i<size; i++)
```

Code

```
1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int l=0, r=0;
5         int res = -99999;
6         int sum = 0;
7         while (r < nums.size()) {
8             while (sum < 0) {
9                 sum -= nums[l];
10                l++;
11            }
12            sum += nums[r];
13        }
14    }
15 }
```

Testcase Test Result

Case 1 Case 2 Case 3 +

nums =

[-2,1,-3,4,-1,2,1,-5,4]

</> Source

240. Search a 2D Matrix II

Problem List < > ✕ Run Submit ⌚ 📄 Premium

Description Editorial Solutions Accepted ✕ Submissions

All Submissions

Accepted 130 / 130 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:01

Editorial Solution

Runtime 52 ms | Beats 63.24% @ Memory 18.65 MB | Beats 67.47%

Analyze Complexity

Code | C++

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = matrix.size(), col = matrix[0].size(), i = 0, j = col - 1;
        while(i < row && j >= 0){
            if(matrix[i][j] == target) return true;
            else if(matrix[i][j] > target) j--;
            else i++;
        }
        return false;
    }
};
```

Saved Ln 1, Col 1

Testcase Test Result

372. Super Pow

Problem List < > ✕ Run Submit ⌚ 📄 Premium

Description Editorial Solutions Accepted ✕ Submissions

All Submissions

Accepted 57 / 57 testcases passed

Jaideep Singh submitted at Feb 05, 2025 16:03

Solution

Runtime 5 ms | Beats 11.42% @ Memory 15.27 MB | Beats 52.19%

Analyze Complexity

Code | C++

```
class Solution {
public:
    const int MOD = 1337;

    // Calculate (a^k) % 1337
    int modPow(int a, int k) {
        a %= MOD;
        int result = 1;
        for (int i = 0; i < k; ++i) {
            result = (result * a) % MOD;
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        if (b.empty()) return 1;

        // Get the last digit in the vector b
        int lastDigit = b.back();
        b.pop_back();

        // Recursively calculate superPow(a, b) and mod for current digit
        int part1 = modPow(superPow(a, b), 10);
        int part2 = modPow(a, lastDigit);

        return (part1 * part2) % MOD;
    }
};
```

Saved Ln 1, Col 1