



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Assignment

Student Name: Khushal

UID: 22CS13927

Branch: CSE

Section/Group: 605-B

Semester: 5

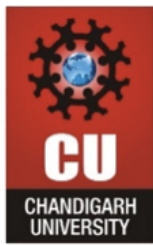
Date of Performance: 5 /02/25

Subject Name: AP

Subject Code: 22CSP-351

Q.1 1763. Longest Nice Substring

```
class Solution {  
  
public:  
  
    bool isNice(const string& str) {  
  
        unordered_set<char> charSet(str.begin(), str.end());  
  
        for (char ch : str) {  
  
            if (charSet.count(tolower(ch)) == 0 || charSet.count(toupper(ch)) == 0) {  
  
                return false;  
  
            }  
  
        }  
  
        return true;  
  
    }  
  
    string longestNiceSubstring(string s) {  
  
        int maxLength = 0;  
  
        string result = "";
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
for (int i = 0; i < s.length(); ++i) {  
    for (int j = i + 1; j <= s.length(); ++j) {  
        string substring = s.substr(i, j - i);  
        if (isNice(substring) && substring.length() > maxLength) {  
            maxLength = substring.length();  
            result = substring;  
        }  
    }  
}  
return result;  
}  
};
```

Output :

← All Submissions

Accepted 73 / 73 testcases passed

codexkhushal submitted at Feb 05, 2025 11:57

Solution

Runtime ⓘ

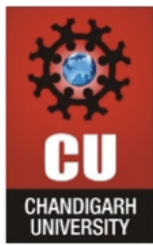
19 ms | Beats **31.34%**

[Analyze Complexity](#)

Memory

14.54 MB | Beats **28.42%**

[Analyze Complexity](#)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q.2 [190. Reverse Bits](#)

```
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); // Shift result to the left and add the last bit of n  
  
            n >>= 1; // Shift n to the right to process the next bit  
  
        }  
  
        return result;  
  
    }  
};
```

[← All Submissions](#)

Accepted 600 / 600 testcases passed

[Solution](#)

codexkh... submitted at Feb 05, 2025 12:11

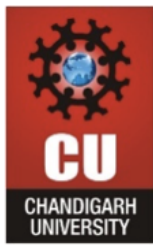
Runtime

0 ms | Beats 100.00% 🏆

[Analyze Complexity](#)

Memory

7.89 MB | Beats 29.58%



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q.3 191. Number of 1 Bits

```
class Solution {  
  
public:  
  
    int hammingWeight(uint32_t n) {  
  
        int res = 0;  
  
        for (int i = 0; i < 32; i++) {  
  
            if ((n >> i) & 1) {  
  
                res += 1;  
  
            }  
  
        }  
  
        return res;  
  
    }  
};
```

← All Submissions

Accepted 598 / 598 testcases passed

Solution

codexkh... submitted at Feb 05, 2025 12:13

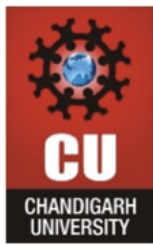
Runtime ⓘ

0 ms | Beats **100.00%** 🏆

🔮 [Analyze Complexity](#)

Memory

8.12 MB | Beats **80.68%** 🏆



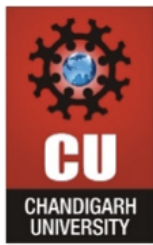
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q.4 [53. Maximum Subarray](#)


```
class Solution {  
  
public:  
  
    int maxSubArray(vector<int>& nums) {  
  
        int maxSum = INT_MIN;  
  
        int currentSum = 0;  
  
        for (int i = 0; i < nums.size(); i++) {  
  
            currentSum += nums[i];  
  
            if (currentSum > maxSum) {  
  
                maxSum = currentSum;  
  
            }  
  
            if (currentSum < 0) {  
  
                currentSum = 0;  
  
            }  
  
        }  
  
        return maxSum;  
  
    }  
};
```

Output






DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Discover. Learn. Empower.

← All Submissions 


Accepted 210 / 210 testcases passed


 **codexkh...** submitted at Feb 05, 2025 12:15


  **Solution**


 **Runtime** 

4 ms | Beats **12.44%**

 [Analyze Complexity](#)

 **Memory**

71.62 MB | Beats **80.58%** 

 [Analyze Complexity](#)

100%

Q.5 [240. Search a 2D Matrix II](#)

```
class Solution {  
public:  
    bool searchMatrix(vector<vector<int>>& matrix, int target) {  
        int r = 0;  
        int c = matrix[0].size()-1;  
  
        while(r<matrix.size() && c < matrix[0].size()){  
  
            if(target == matrix[r][c]){  
                return true;  
            }  
        }  
    }  
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
}  
  
else if(target > matrix[r][c]){  
  
    r++;  
  
}  
  
else{  
  
    c--;  
  
}  
  
  
}  
  
return false;  
  
}  
  
};
```

Output :

← All Submissions

Accepted 130 / 130 testcases passed

Solution

codexkh... submitted at Feb 05, 2025 12:17

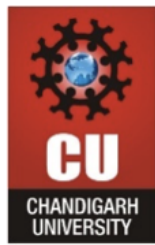
⌚ Runtime ⓘ

44 ms | Beats **88.22%** 🌿

✦ Analyze Complexity

⚙️ Memory

18.51 MB | Beats **91.95%** 🌿



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q.6 372.[Super Pow](#)

```
class Solution {  
  
private:  
  
    int solve(int base, int power, int mod) {  
  
        int ans = 1;  
  
        while (power > 0) {  
  
            if (power & 1) {  
  
                ans = (ans * base) % mod;  
  
            }  
  
            base = (base * base) % mod;  
  
            power >>= 1;  
  
        }  
  
        return ans;  
  
    }  
  
}
```

```
public:  
  
    int superPow(int a, vector<int>& b) {  
  
        a%=1337;  
  
        int n = b.size();  
  
        int m = 1140;  
  
        int expi = 0;  
  
        for(int i : b){  
  
            expi = (expi*10+i)%m;  
  
        }  
  
    }  
  
}
```





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING


Discover. Learn. Empower.



```
}  
  
if (expi == 0) {  
    expi = m;  
}  
  
return solve(a,expi,1337);  
  
}  
  
};
```


Output :


← All Submissions 


Accepted 57 / 57 testcases passed

 **codexkhushal** submitted at Feb 05, 2025 12:20 [Solution](#)

 **Runtime** 

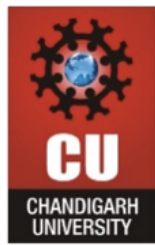
0 ms | Beats **100.00%** 

 [Analyze Complexity](#)

 **Memory**

15.30 MB | Beats **18.19%**

75%



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q.7. [932. Beautiful Array](#)

```
vector<int> beautifulArray(int N) {  
    vector<int> res = {1};  
    while (res.size() < N) {  
        vector<int> tmp;  
        for (int i : res) if (i * 2 - 1 <= N) tmp.push_back(i * 2 - 1);  
        for (int i : res) if (i * 2 <= N) tmp.push_back(i * 2);  
        res = tmp;  
    }  
    return res;  
}
```

Output:

← All Submissions

Accepted 38 / 38 testcases passed

codexkh... submitted at Feb 05, 2025 12:23

Solution

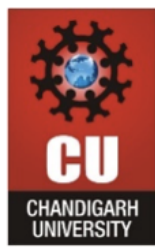
Runtime

0 ms | Beats **100.00%**

Analyze Complexity

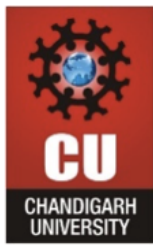
Memory

10.11 MB | Beats **37.82%**



Q.8 [218. The Skyline Problem](#)

```
vector<pair<int, int>> getSkyline(vector<vector<int>>& buildings) {  
  
    // use walls to record buildings; left wall is an insertion event, and right wall is a deletion event  
  
    vector<pair<int, int>> walls, ans;           // first: x, second: height  
  
    for (auto b : buildings) {  
  
        // push in left / right walls  
  
        // let left wall has negative height to ensure left wall goes to multiset first if with same 'x' as right  
wall  
        walls.push_back(make_pair(b[0], -b[2]));  
        walls.push_back(make_pair(b[1], b[2]));  
  
    }  
  
    sort(walls.begin(), walls.end());           // sort walls  
  
  
    multiset<int> leftWallHeights = {0};        // keep left wall heights sorted; dummy '0' for  
convenience  
  
    int top = 0;                                // current max height among leftWallHeights  
  
    for (auto w : walls) {  
  
        if (w.second < 0) {                     // it's a left wall, insert the height  
  
            leftWallHeights.insert(-w.second);  
  
        } else {                                // it's a right wall, delete the height  
  
            leftWallHeights.erase(leftWallHeights.find(w.second));  
  
        }  
    }  
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
if (*leftWallHeights.rbegin() != top) {    // mark a skyline point if top changes

    ans.push_back(make_pair(w.first, top = *leftWallHeights.rbegin()));

}

}

return ans;

};
```

← All Submissions

Accepted 44 / 44 testcases passed

codexkh... submitted at Feb 05, 2025 12:26

Solution

Runtime

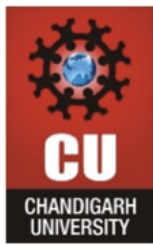
28 ms | Beats **31.02%**

[Analyze Complexity](#)

Memory

30.10 MB | Beats **40.10%**

Output :



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
class Solution {  
  
public:  
  
    int reversePairs(vector<int>& nums) {  
  
        int n = nums.size();  
  
        long long reversePairsCount = 0;  
  
        for(int i=0; i<n-1; i++){  
  
            for(int j=i+1; j<n; j++){  
  
                if(nums[i] > 2*(long long)nums[j]){  
  
                    reversePairsCount++;  
  
                }  
  
            }  
  
        }  
  
        return reversePairsCount;  
  
    }  
  
};
```

Output :

← All Submissions [Link](#)

Accepted 600 / 600 testcases passed

codexkh... submitted at Feb 05, 2025 12:11 [Solution](#)

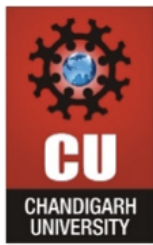
Runtime

0 ms | Beats **100.00%**

[Analyze Complexity](#)

Memory

7.89 MB | Beats **29.58%**



Q.10 [2407. Longest Increasing Subsequence II](#)

```
class Solution {  
  
public:  
  
    vector<int>tree;  
  
    void update(int node,int st,int end,int i,int val){  
  
        if(st==end){  
  
            tree[node]=max(tree[node],val);  
  
            return;  
  
        }  
  
        int mid=(st+end)/2;  
  
        if(i<=mid){  
  
            update(node*2,st,mid,i,val);  
  
        }else{  
  
            update(node*2+1,mid+1,end,i,val);  
  
        }  
  
        tree[node]=max(tree[node*2],tree[node*2+1]);  
  
    }  
  
    int query(int node,int st,int end,int x,int y){  
  
        if(x>end || y<st) return -1e9;  
  
        if(st>=x && end<=y){  
  
            return tree[node];  
  
        }  
  
    }  
  
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
    }

    int mid=(st+end)/2;

    int left=query(2*node,st,mid,x,y);

    int right=query(2*node+1,mid+1,end,x,y);

    return max(left,right);

}

int lengthOfLIS(vector<int>& nums, int k) {

    int n=nums.size();

    if(n==1) return 1;

    int m=*max_element(nums.begin(),nums.end());

    tree.clear();

    tree.resize(4*m+10);

    for(int i=n-1;i>=0;i--){

        int l=nums[i]+1,r=min(nums[i]+k,m);

        int x=query(1,0,m,l,r);

        if(x==-1e9) x=0;

        update(1,0,m,nums[i],x+1);

    }

    return tree[1];

}

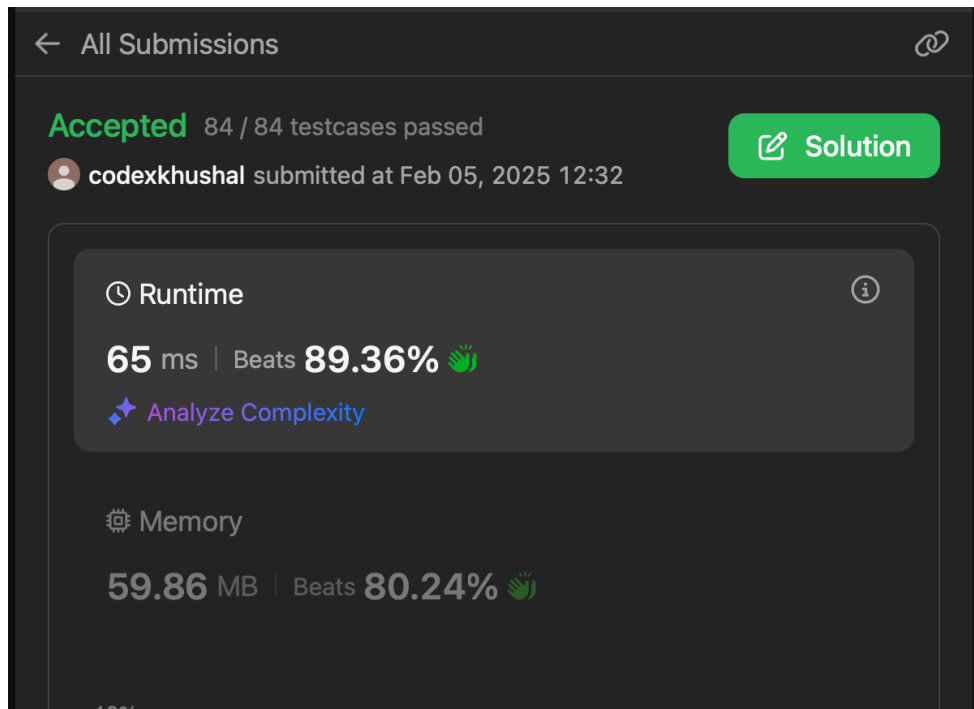
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Output :



Q.11 [88. Merge Sorted Array](#)

```
class Solution {
```

```
public:
```

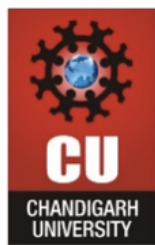
```
void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
```

```
    for(int j = 0, i = m ; j < n ; j++){
```

```
        nums1[i] = nums2[j];
```

```
        i++;
```

```
    }
```

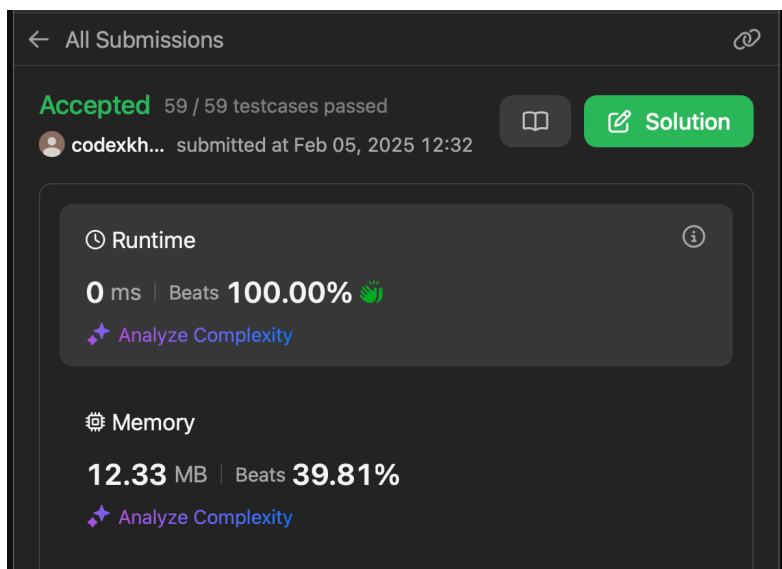



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        sort(nums1.begin(),nums1.end());  
  
    }  
  
};
```

Output :



Q.12 [278. First Bad Version](#)

// The API isBadVersion is defined for you.

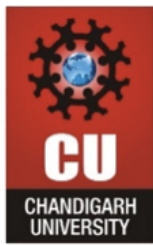
```
bool isBadVersion(int version);
```

```
class Solution {
```

```
public:
```

```
    int firstBadVersion(int n) {
```

```
        int low=1;
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
int high=n;
```

```
while(low<=high){
```

```
    int mid=low+(high-low)/2;
```

```
    int version=isBadVersion(mid);
```

```
    if(version==true){
```

```
        high=mid-1;
```

```
    }
```

```
    else{
```

```
        low=mid+1;
```

```
    }
```

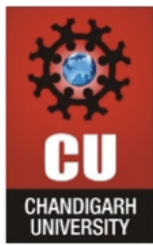
```
}
```

```
return low;
```

```
}
```


```
};
```

Output :






DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



Discover. Learn. Empower.


← All Submissions 


Accepted 24 / 24 testcases passed


  **Solution**

 **codexkh...** submitted at Feb 05, 2025 12:34

 **Runtime** 

2 ms | Beats **54.93%** 

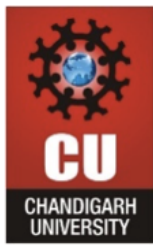
 [Analyze Complexity](#)

 **Memory**

7.96 MB | Beats **38.30%**

Q.13 [75. Sort Colors](#)

```
class Solution {  
  
public:  
  
    void sortColors(vector<int>& nums) {  
  
        int n = nums.size();  
  
        int i = 0 ;  
  
        int j = 0 ;  
  
        int k = n-1;  
  
        while(j<=k){
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
if(nums[j]==1){  
    j++;  
}else if(nums[j]==2){  
    swap(nums[j],nums[k]);  
    k--;  
}  
else{//nums[j]==0  
    swap(nums[j] , nums[i]);  
    i++;  
    j++;  
}  
}  
}  
};
```

Output :

← All Submissions

Accepted 88 / 88 testcases passed

codexkh... submitted at Feb 05, 2025 12:35

[Solution](#)

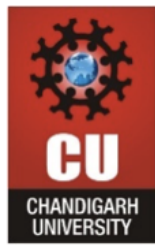
Runtime

0 ms | Beats 100.00%

[Analyze Complexity](#)

Memory

11.54 MB | Beats 66.83%



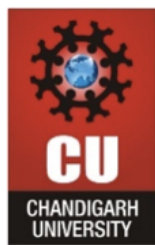
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Q.14 [347. Top K Frequent Elements](#)

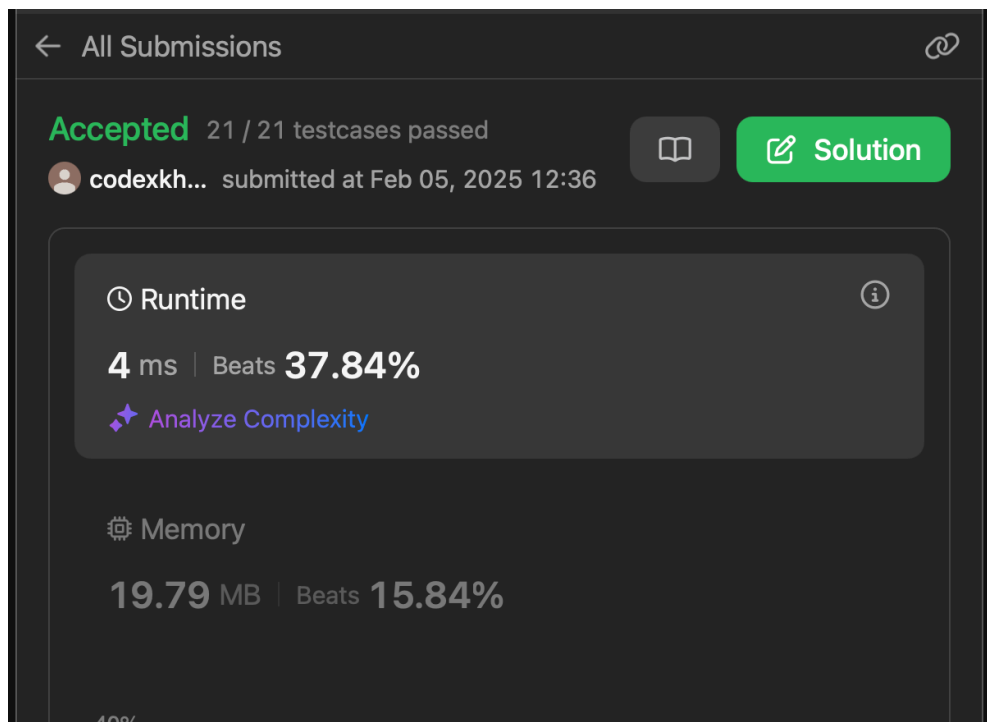
```
class Solution {  
  
public:  
  
    vector<int> topKFrequent(vector<int>& nums, int k) {  
  
        int n = nums.size();  
  
        unordered_map<int, int> map;  
  
        vector<int> ans;  
  
        for (int &x : nums) map[x]++;  
  
        vector<vector<int>> arr(n + 1);  
  
        for (auto [a, b] : map) arr[b].push_back(a);  
  
        for (int i = n; i > 0; i--) {  
  
            for (int &x : arr[i]) {  
  
                if (ans.size() == k) return ans;  
  
                ans.push_back(x);  
  
            }  
  
        }  
  
        return ans;  
  
    }  
  
};
```

Output :



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



Q.15 [215. Kth Largest Element in an Array](#)

```
class Solution {  
  
public:  
  
    int findKthLargest(vector<int>& nums, int k) {  
  
        sort(nums.begin(), nums.end());  
  
        return nums[nums.size() - k];  
  
    }  
  
};
```

Output :



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

← All Submissions



Accepted 42 / 42 testcases passed



codexkh... submitted at Feb 05, 2025 12:36



Solution

🕒 Runtime

23 ms | Beats **79.92%** 🌱

💻 Memory

59.24 MB | Beats **83.52%** 🌱

🌟 [Analyze Complexity](#)