

## 108. Convert Sorted Array to Binary Search Tree

Accepted

All Submissions

Runtime  
13 ms | Beats 7.42%

Analyze Complexity

Memory  
21.96 MB | Beats 98.98%

60%  
40%  
20%

```
10 * };
11 */
12 class Solution {
13 public:
14     TreeNode* sortedArrayToBST(vector<int>& nums) {
15         return sortedArrayToBSTHelper(nums,0,nums.size()-1);
16     }
17 private:
18     TreeNode* sortedArrayToBSTHelper(vector<int>& nums, int left,int right)
19     {
20         if(left>right){
21             return NULL;
22         }
23         int mid=left+(right-left)/2;
24         TreeNode* root= new TreeNode(nums[mid]);
25         root->left=sortedArrayToBSTHelper(nums,left,mid-1);
26         root->right=sortedArrayToBSTHelper(nums,mid+1,right);
27         return root;
28     };
29 }
```

## 191. Number of 1 Bits

Accepted

All Submissions

Runtime  
0 ms | Beats 100.00%

Analyze Complexity

Memory  
8.27 MB | Beats 47.48%

100%  
50%  
0%

1ms 2ms 3ms 4ms

1.51% of solutions used 2 ms of runtime

```
1 class Solution {
2 public:
3     int hammingWeight(int n) {
4         int count=0;
5         for(int i=0;i<32;i++){
6             int temp=n>>i;
7             int a=temp&1;
8
9             if(a==1){
10                 count++;
11             }
12         }
13         return count;
14     }
15 };
16 ;;
```

## 912. Sort an Array

Accepted 21 / 21 testcases passed

Rakshita... submitted at Mar 22, 2025 22:10

Solution

Runtime

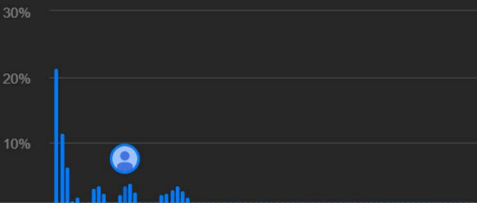
375 ms | Beats 41.37%

Analyze Complexity

Memory

185.82 MB | Beats 22.17%

Analyze Complexity



Code

C++ Auto

```
1 class Solution {
2 public:
3     void merge(vector<int>& nums, int left, int mid, int right) {
4         int n1 = mid - left + 1;
5         int n2 = right - mid;
6
7         vector<int> leftArr(n1), rightArr(n2);
8
9         for (int i = 0; i < n1; i++) leftArr[i] = nums[left + i];
10        for (int i = 0; i < n2; i++) rightArr[i] = nums[mid + 1 + i];
11
12        int i = 0, j = 0, k = left;
13
14        while (i < n1 && j < n2) {
15            if (leftArr[i] <= rightArr[j]) {
16                nums[k++] = leftArr[i++];
17            } else {
18                nums[k++] = rightArr[j++];
19            }
20        }
21    }
```

Saved

Test Result | Testcase

## 53. Maximum Subarray

Accepted 210 / 210 testcases passed

Rakshita... submitted at Oct 20, 2024 22:13

Solution

Runtime

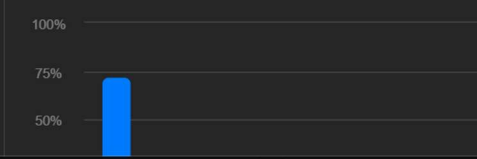
1 ms | Beats 28.10%

Analyze Complexity

Memory

70.32 MB | Beats 99.97%

Analyze Complexity



Code

C++ Auto

```
1 class Solution {
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int n = nums.size();
5         int i = 1, sum = nums[0], max = 0;
6         while (i < n) {
7             sum += nums[i];
8             if (sum > max) {
9                 max = sum;
10            }
11            else {
12                sum = 0;
13            }
14            i++;
15        }
16        return max;
17    }
18 };
```

Saved

Test Result | Testcase

## 932. Beautiful Array

Editorial Solutions Submissions Accepted X

All Submissions

Accepted 38 / 38 testcases passed

Rakshita... submitted at Mar 22, 2025 22:13

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

10.08 MB | Beats 59.10%

75%

50%

25%

Code

C++ Auto

```
1 class Solution {
2 public:
3     vector<int> beautifulArray(int n) {
4         vector<int> result = {1};
5         while (result.size() < n) {
6             vector<int> temp;
7             for (int num : result) {
8                 if (2 * num - 1 <= n) temp.push_back(2 * num - 1);
9             }
10            for (int num : result) {
11                if (2 * num <= n) temp.push_back(2 * num);
12            }
13            result = temp;
14        }
15        return result;
16    }
17 };
```

Saved

Test Result Testcase

## 372. Super Pow

Description Accepted X Editorial Solutions Submissions

All Submissions

Accepted 57 / 57 testcases passed

Rakshita Thakur submitted at Mar 22, 2025 22:16

Solution

Runtime

4 ms | Beats 18.40%

Analyze Complexity

Memory

15.32 MB | Beats 14.66%

75%

50%

25%

Code

C++ Auto

```
1 class Solution {
2 public:
3     const int MOD = 1337;
4     int modPow(int x, int y) {
5         int result = 1;
6         x %= MOD;
7         while (y > 0) {
8             if (y % 2 == 1) result = (result * x) % MOD;
9             x = (x * x) % MOD;
10            y /= 2;
11        }
12        return result;
13    }
14
15    int superPow(int a, vector<int>& b) {
16        a %= MOD;
17        int result = 1;
18        for (int digit : b) {
19            result = (result * modPow(a, digit)) % MOD;
20        }
21        return result;
22    }
23 };
```

Saved

Test Result Testcase

Accepted Runtime: 0 ms