

EXP 6 SOLUTIONS

[1] [Convert Sorted Array to Binary Search Tree](#)

The screenshot displays a coding interface for the problem "108. Convert Sorted Array to Binary Search Tree". The problem description states: "Given an integer array `nums` where the elements are sorted in **ascending order**, convert it to a **height-balanced** binary search tree." An example shows a tree with root 0, left child -3, and right child 9.

```
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int> &nums)
    {
        return treeConstructor(nums, 0, nums.size() - 1);
    }

    TreeNode* treeConstructor(vector<int> nums, int left, int right)
    {
        if (left > right)
            return nullptr;

        int mid = (left + right) / 2;
        TreeNode* node = new TreeNode();

        node->val = nums[mid];

        node->left = treeConstructor(nums, left, mid - 1);
        node->right = treeConstructor(nums, mid + 1, right);

        return node;
    }
};
```

[2] [Number of 1 Bits](#)

The screenshot displays a coding interface for the problem "191. Number of 1 Bits". The problem description states: "Given a positive integer `n`, write a function that returns the number of **set bits** in its binary representation (also known as the **Hamming weight**)." Example 1 shows input `n = 11` and output `3`, with an explanation that the binary string `1011` has three set bits.

```
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        for(int i = 31; i >= 0; i--){
            if(((n >> i) & 1) == 1)
                count++;
        }
        return count;
    }
};
```

[3] [Sort an Array](#)

Description

Editorial

Solutions

Submissions

932. Beautiful Array

Solved

Medium Topics Companies

An array `nums` of length `n` is **beautiful** if:

- `nums` is a permutation of the integers in the range `[1, n]`.
- For every $0 \leq i < j < n$, there is no index `k` with $i < k < j$ where $2 * \text{nums}[k] == \text{nums}[i] + \text{nums}[j]$.

Given the integer `n`, return *any beautiful array* `nums` of length `n`. There will be at least one valid answer for the given `n`.

Example 1:

Input: `n = 4`
Output: `[2, 1, 4, 3]`

Code

C++ Auto

```

1 class Solution {
2 public:
3     static bool comp(const int &a, const int &b){
4         int mask = 1;
5         while(true)
6             if((a&mask) == (b&mask)) mask = mask<<1;
7         else return (a&mask) > (b&mask);
8     }
9
10    vector<int> beautifulArray(int n) {
11        vector<int> answer;
12        while(n) answer.push_back(n--);
13
14        sort(answer.begin(), answer.end(), comp);
15
16        return answer;
17    }
18 };

```

Testcase Test Result

[6] [Super Pow](#)

Problem List

Description

Editorial

Solutions

Submissions

372. Super Pow

Solved

Medium Topics Companies

Your task is to calculate $a^b \bmod 1337$ where `a` is a positive integer and `b` is an extremely large positive integer given in the form of an array.

Example 1:

Input: `a = 2, b = [3]`
Output: `8`

Example 2:

Input: `a = 2, b = [1,0]`
Output: `1024`

Example 3:

Input: `a = 1, b = [4,3,3,8,5,2]`
Output: `1`

Code

C++ Auto

```

1 class Solution {
2 private:
3     int solve(int base, int power, int mod) {
4         int ans = 1;
5         while (power > 0) {
6             if (power & 1) {
7                 ans = (ans * base) % mod;
8             }
9             base = (base * base) % mod;
10            power >>= 1;
11        }
12        return ans;
13    }
14
15 public:
16    int superPow(int a, vector<int>& b) {
17        a%=1337;
18        int n = b.size();
19        int m = 1140;
20        int exp1 = 0;
21        for(int i : b){
22            exp1 = (exp1*10+i)%m;

```

Testcase Test Result

[7] [The Skyline Problem](#)

Problem List

218. The Skyline Problem

Solved

Hard

Topics

Companies

A city's **skyline** is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Given the locations and heights of all the buildings, return the **skyline** formed by these buildings collectively.

The geometric information of each building is given in the array `buildings` where `buildings[i] = [lefti, righti, heighti]`:

- `lefti` is the x coordinate of the left edge of the `ith` building.
- `righti` is the x coordinate of the right edge of the `ith` building.
- `heighti` is the height of the `ith` building.

6K

31

43 Online

Code

C++

Auto

Ln 1, Col 1

```
1 class Solution {
2 public:
3     vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
4         int edge_idx = 0;
5         vector<pair<int, int>> edges;
6         priority_queue<pair<int, int>> pq;
7         vector<vector<int>> skyline;
8
9         for (int i = 0; i < buildings.size(); ++i) {
10             const auto &b = buildings[i];
11             edges.emplace_back(b[0], b[1]);
12             edges.emplace_back(b[1], b[1]);
13         }
14
15         sort(edges.begin(), edges.end());
16
17         while (edge_idx < edges.size()) {
18             int curr_height;
19             const auto &[curr_x, _] = edges[edge_idx];
20             while (edge_idx < edges.size() &&
21                  curr_x == edges[edge_idx].first) {
22                 const auto &[, building_idx] = edges[edge_idx];
```

Testcase

Test Result