1. https://leetcode.com/problems/kth-smallest-element-in-a-bst/description/

```
class Solution {
public:
    // priority_queue<int, vector<int>, greater<int>> pq;
    set<int> s;
    void preorder(TreeNode *root){
        if(root == NULL) return;
        // pq.push(root->val);
        s.insert(root->val);
        preorder(root->left);
        preorder(root->right);
    }
    int kthSmallest(TreeNode* root, int k) {
        preorder(root);
        auto it = s.begin();
        advance(it, k-1);
        return *it;
    }
};
```

2. https://leetcode.com/problems/top-k-frequent-elements/description/

```
class cmp{
public:
    bool operator()(pair<int, int> &a, pair<int, int> &b){
        if(a.second == b.second) return a.first < b.first;
        return a.second < b.second;
    }

};
class Solution {
public:
    vector<int> topKFrequent(vector<int>& nums, int k) {
        map<int, int> mp;
        for(int x: nums) mp[x]++;

        priority_queue<pair<int,int>,vector<pair<int, int>>, cmp> pq;

        for(auto x : mp) pq.push(x);

        vector<int> ans;
```

```
        while(k--) ans.push_back(pq.top().first), pq.pop();
        return ans;
    }
};
```

3. https://leetcode.com/problems/contains-duplicate/description/

```
class Solution {
public:
    bool containsDuplicate(vector<int>& nums) {
        set<int> s(nums.begin(), nums.end());
        // for(int x: nums) s.insert(x);
        return s.size() != nums.size();
    }
};
```

4. https://codeforces.com/problemset/problem/469/A

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int level, x;
    set<int> s;
    cin >> level;

    int p, q;
    cin >> p;
    while(p--)cin>>x, s.insert(x);
    cin >> q;
    while(q--) cin>>x, s.insert(x);

    cout << ((s.size() == level) ? "I become the guy." : "Oh, my keyboard!") << endl;
}
```

5. https://codeforces.com/problemset/problem/22/A

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int n, x;
    set<int> s;
    cin >> n;
    for(int i = 0; i < n; i++)cin >> x, s.insert(x);
    if(s.size() < 2)
```

```
    {
        cout << "NO"; return 0;
    }
    auto it = s.begin();
    advance(it, 1);
    cout << *it << endl;

}
```