Competitive Coding

Name : Harsh

22BCE0782

Que: Build a Staircase

#include <bits/stdc++.h>

using namespace std;

int main() {

// your code goes here

int t;

cin>>t;

while (t--){

int N;

cin >> N;

vector<int> blocks(N);

for (int i = 0; i < N; i++) {

cin >> blocks[i];

}

sort(blocks.begin(), blocks.end());

int left = 0, right = N - 1;

bool possible = true;

while (left <= right) {

int sum = blocks[left] + blocks[right];

if (sum == blocks[N - 1]) {

left++;

right--;

} else if (sum < blocks[N - 1]) {

left++;

} else {

possible = false;

break;

}

}

if (possible) {

cout << "Yes" << endl;

} else {

cout << "No" << endl;

}

}

return 0;

}



Que :

#include <bits/stdc++.h>

using namespace std;

int main() {

int t;

cin >> t;

while (t--) {

long long N, K;

cin >> N >> K;

long long count = 0;

while (K > N) {

long long largest\_power\_of\_2 = 1LL << static\_cast<int>(log2(K - N));

K -= largest\_power\_of\_2;

count++;

}

cout << count << endl;

}

return 0;

}



Que : BitwiseXOR

#include <bits/stdc++.h>

using namespace std;

int main() {

// your code goes here

int t;

cin>>t;

while (t--){

int size, num, reqSum, reqXOR, ans;

ans = 0;

cin>>size;

vector<int> v;

for (int i=0; i<size; i++){

cin>>num;

v.push\_back(num);

}

sort(v.begin(), v.end());

cin>>reqXOR>>reqSum;

int i = 0;

int j = size - 1;

while (i<j){

if ((v[i]+v[j]) == reqSum){

if ((v[i]^v[j]) == reqXOR){

ans++;

}

// if (v[i] == v[i+1]) i++;

// else if (v[j] == v[j-1]) j--;

// else {

// i++;

// j--;

// }

i++;

j--;

}

else if ((v[i]+v[j]) > reqSum) j--;

else i++;

}

cout<<ans;

}

return 0;

}

