1624C - Division by Two and Permutation

Idea: **MikeMirzayanov**
Tutorial

1624C - Division by Two and Permutation

Let's sort the array a in descending order of the values of its elements. Then let's create a logical array $used$, where $used[i]$ will have the value `true` if we already got element i of the permutation we are looking for, and the value `false` otherwise.

We loop through the elements of the array a and assign $x = a_i$. We'll divide x by 2 as long as it exceeds n or as long as $used[x]$ is `true`.

- If it turns out that $x = 0$, then all the numbers that could be obtained from a_i have already been obtained before. Since each element of the array a must produce a new value from 1 to n, the answer cannot be constructed — output `NO`.
- Otherwise, assign $used[x]$ a value of `true` — this means that the number x, which is an element of the permutation, we will get exactly from the original number a_i.

After processing all elements of the array a we can output `YES`.

Solution

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

void solve(){
    int n;
    cin >> n;
    vector<int>a(n), used(n + 1, false);
    for(auto &i : a) cin >> i;
    sort(a.begin(), a.end(), [] (int a, int b) {
        return a > b;
```

```
    });
    bool ok = true;
    for(auto &i : a){
        int x = i;
        while(x > n or used[x])  x /= 2;
        if(x > 0) used[x] = true;
        else ok = false;
    }
    cout << (ok ? "YES" : "NO") << '\n';

}

int main(){
    ios_base :: sync_with_stdio(false);
    cin.tie(nullptr);
    int t;
    cin >> t;
    while(t--){
        solve();
    }
    return 0;
}
```