列情况找规律的题，

1 2 4 8 16，2的n次方，可以拼凑出1-n的任意数字

这题我的思路没错，但是漏了些关键点，比如目标序列为{1，1，2，5，7}，我的程序从0到n-1遍历目标序列，初始cnt设置为1，导致遍历到元素5时，cnt为5，本来应该输出No，最后却因为f[i] <= cnt而输出了yes

将cnt改为0，又会导致第一项1 > cnt而输出no，当时太着急了，一直找chatgpt没有自己好好看，导致这道题和后一道每过，不过这也是好事，毕竟如果我打了七道题比其他牛人还要多的话我一定会飘飘然认不清自己的吧

题解思路：最终要改为cnt = 1，然后从n = 1开始遍历，再加一个特判，检查第一项是不是1

至于hard版本·，数据范围上到2e5项，大小2e5，爆int，换成long long就完事

#include <iostream>

#include <cstring>

#include <cstdio>

#include <algorithm>

using namespace std;

const int N = 2e5 + 10;

typedef long long ll;

int main()

{

int t;

cin >> t;

while(t--){

ll n;

cin >> n;

ll f[N];

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

cin >> f[i];

}

sort(f,f+n);

bool flag = true;

ll cnt = 1;

if(f[0]!=cnt){

flag = false;

}

for(int i = 1;i < n;i++){

if(f[i] <= cnt) {

cnt+= f[i];

}

else {

flag = false;

break;

}

}

if(flag){

cout << "YES" << endl;

}

else cout <<"NO" << endl;

}

return 0;

}