Subset Sum Problem

subset(i,j)表示S中前i个元素的子集和等于j的情况，则

- 若S[i] > j，则S[i]不在子集s中。
- 若S[i] <= j, 则有以下两种情况：一种情况是S[i]不在子集s中，则subset(i, j) = subset(i-1, j); 一种情况是S[i]在子集s中，则subset(i, j)= subset(i-1, j-S[i]).

对于i=0,1,2,...,n,有subset(i, 0)=True, 对于j=1,2,...,M, 有subset(0, j)=False.

首先判断集合中所有元素的和sum

1. 如果 sum % 2 ≠ 0，那么该集合是一定不能划分成两个和相等的子集
2. 如果 sum % 2 == 0，那么该集合还需要再进行判断（利用上面的方法），不过只需要计算 subset(n, sum / 2)

```
#include <bits/stdc++.h>

using namespace std;

bool isSubsetSum(vector<int> set, int n, int sum) {
    bool subset[n + 1][sum + 1];

    for (int i = 0; i <= n; i++) {
        subset[i][0] = true;
    }

    for (int i = 1; i <= sum; i++) {
        subset[0][i] = false;
    }

    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= sum; j++) {
            if (j < set[i - 1])
                subset[i][j] = subset[i - 1][j];
            if (j >= set[i - 1])
                subset[i][j] = subset[i - 1][j] ||
                               subset[i - 1][j - set[i - 1]];
        }
    }

    return subset[n][sum];
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
```

```
        vector<int> set;
        for (int i = 0; i < N; ++i) {
            int num;
            scanf("%d", &num);
            set.push_back(num);
        }
        int sum = 0;
        for (int s: set)
            sum += s;
        if (sum % 2 != 0) printf("NO\n");
        else isSubsetSum(set, N, sum / 2) ? printf("YES\n") : printf("NO\n");

    }
    return 0;
}
```