**USACO ORZ**

Like everyone, cows enjoy variety. Their current fancy is new shapes for pastures. The old rectangular shapes are out of favor; new geometries are the favorite.   
I. M. Hei, the lead cow pasture architect, is in charge of creating a triangular pasture surrounded by nice white fence rails. She is supplied with N fence segments and must arrange them into a triangular pasture. Ms. Hei must use all the rails to create three sides of non-zero length. Calculating the number of different kinds of pastures, she can build that enclosed with all fence segments.   
Two pastures look different if at least one side of both pastures has different lengths, and each pasture should not be degeneration.

**Input**

The first line is an integer T(T<=15) indicating the number of test cases.   
The first line of each test case contains an integer N. (1 <= N <= 15)   
The next line contains N integers li indicating the length of each fence segment. (1 <= li <= 10000)

**Output**

For each test case, output one integer indicating the number of different pastures.

**Sample Input**

1

3

2 3 4

**Sample Output**

1

题目大意：给定一些一定长度的线段，要求全部利用这些线段能拼成多少种三角形（如果两个三角形至少有一条边长度不等那么二者视为两种）

解题方法：dfs+set判重

解题思路：O（3^n） dfs，对于所有满足情况的三角形加入set中，这样就去重了，最后set的大小就是答案。注意题意是全部利用这些线段能拼成多少种三角形，所以才会有ans

附代码

#include<bits/stdc++.h>

using namespace std;

set<long long>s;

int a[20];

int ans,n,t;

void dfs(int cnt,int x,int y)//x,y是两条边，cnt是个计数的，因为要枚举一遍所有的边加在不同的xy两边上

{

int z=ans-x-y;//第三条边

if(cnt>n)

{

if(x<=y&&y<=z&&x+y>z)//特意规定了x,y有序，是为了避免2,1,5和1,2,5的情况

s.insert(x\*10000000+y);//x乘以一个很大的数+y，就是存下xy两条边，xy不同，加出来的值就不同

return;

}

dfs(cnt+1,x+a[cnt],y);

dfs(cnt+1,x,y+a[cnt]);

dfs(cnt+1,x,y);

}

int main()

{

cin>>t;

while(t--)

{

cin>>n;

ans=0;

s.clear();

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

{

cin>>a[i];

ans+=a[i];//ans是所有边的总和

}

dfs(1,0,0);

cout<<s.size()<<endl;

}

return 0;

}

信息对抗1602

郝希烜