

[解题报告]I No Tipping

[Source] <http://202.114.18.202:8080/judge/contest/contest/view.action?cid=6205#problem/I>

[Description]

一块长 L 重 W 的木板，中间位置坐标为 0 ，上面放了一些物体，第 i 个物体坐标 $x[i]$ ，重量 $w[i]$ ，两个支点坐标分别为 $+1.5$ 和 -1.5 。问能不能以一定的顺序将所有的物体拿下来，并维持木板始终不会倾倒。

[Solution]

这题的题意挺复杂的，不过理解清楚了之后，搜索的部分是很简单的。

这道题还用到了了一定的物理知识，对一个支点的转矩就是所有的物体相对于支点的坐标乘重量再求和，转矩大于 0 时木板顺时针转，小于 0 时逆时针转。此题中有两个支点的木板平衡的条件显然是左边支点的转矩大于等于 0 且右边支点的转矩小于等于 0 。注意木板本身可以视作一个放在 0 点，重量 w 的物体。还有为了避免浮点误差可以将所有的坐标都乘以 2 。

搜索的部分直接 `dfs`，枚举还在木板上的物体，如果拿掉之后木板依旧平衡就深搜下一层。搜到所有的物体都拿掉之后输出答案。

注意已经拿掉的物体拿掉的顺序不会对剩下的物体产生影响，所以可以对状态进行记忆化，因为物体最多 20 个，所以可以用一个小于 2^{20} 的二进制数记录每一个物体是否还在，当遇到一个已经搜过的状态时直接 `return`。

[Code]

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<cmath>
#include<algorithm>

using namespace std;
const int N=22;
int x[N],w[N],n,flag;
int v[1050000],res[N];

void dfs(int p,int s1,int s2,int s)
{
    if (s1<0 || s2>0) return;
    if (flag) return;
    if (v[s]) return; else v[s]=1;
    int i;
    if (p>n)
    {
```

```

        for (i=1; i<=n; i++)
            printf("%d %d\n",x[res[i]],w[res[i]]);
        flag=1; return;
    }
    for (i=1; i<=n; i++)
        if ((s&(1<<(i-1)))==0)
        {
            res[p]=i;
            dfs(p+1,s1-w[i]*(x[i]*2+3),s2-w[i]*(x[i]*2-3),s+(1<<(i-1)));
        }
    }

int main()
{
    int l,tw,i,s1,s2,cas=0;
    while (scanf("%d%d%d",&l,&tw,&n)==3)
    {
        if (l==0 && tw==0 && n==0) break;
        s1=tw*3; s2=-tw*3;
        for (i=1; i<=n; i++)
        {
            scanf("%d%d",&x[i],&w[i]);
            s1+=w[i]*(x[i]*2+3);
            s2+=w[i]*(x[i]*2-3);
        }
        memset(v,0,sizeof(v));
        printf("Case %d:\n",++cas);
        flag=0;
        dfs(1,s1,s2,0);
        if (flag==0) printf("Impossible\n");
    }
}

```