题解：经典的dp问题，给出n种类型的长方体石块，给出每种的长、宽、高，每种石块有三种摆放方式，数量不限，可以累加着往上放，问最多能放多高？但是有个限定条件，长和宽必须均小于下面石块的长和宽才能放上去。

思想还是最长下降子序列

因为题意并没有要求是连续最长下降子序列的，并且题目要求严格递减

所以排个序

#include<iostream>

#include<stdio.h>

#include<algorithm>

#include<string.h>

#define INF 1e9+7

using namespace std;

struct node

{

int length,width,height;

}block[500];

int dp[100];

bool cmp(node a,node b)

{

if(a.length==b.length)

return a.width>b.width;

else return a.length>b.length;

}

int main()

{

//freopen("input.txt","r",stdin);

int n,x,y,z;

int o=0;

while(cin>>n && n)

{

o++;

int l=0;

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

{

cin>>x>>y>>z;

block[l].length=x;block[l].width=y;block[l++].height=z;

block[l].length=x;block[l].width=z;block[l++].height=y;

block[l].length=y;block[l].width=x;block[l++].height=z;

block[l].length=y;block[l].width=z;block[l++].height=x;

block[l].length=z;block[l].width=x;block[l++].height=y;

block[l].length=z;block[l].width=y;block[l++].height=x;

}

sort(block,block+l,cmp);

memset(dp,0,sizeof(dp));

dp[0]=block[0].height;

int ans(0);

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

{

dp[i]=block[i].height;

for(int j=0;j<i;j++)

{

if(block[i].length<block[j].length && block[i].width<block[j].width)

dp[i]=max(dp[i],dp[j]+block[i].height);

}

ans=max(ans,dp[i]);

}

cout<<"Case "<<o<<": maximum height = ";

cout<<ans<<"\n";

}

return 0;

}