计算机算法设计与分析

**学院：计算机科学与工程学院 班级：计算机172班 姓名：李源 学号：201706401239**

**一、实验题目**

多维背包问题。  
有11个数据文件。

第一个数据文件是mknap1。这个数据文件包含7个测试问题，它们是。C.Petersen“计算经验”中的测试问题。将Balas算法的变体应用于选择。研发项目“管理科学”13(9)(1967)736-750。要解决的问题是：最大和{j=1，.，n}p(J)x(J)。st sum{j=1，.，n}r(i，j)x(J)<=b(I)i=1，.，m。X(J)=0或1。

此数据文件的格式为：  
测试问题数(K)。  
然后对于每个测试问题k(k=1，.，K)依次：  
变量数(N)，约束数(M)，最优。  
解决方案值(如果不可用，则为零)。  
系数p(J)；j=1，…，n。  
对于每个约束i(i=1，.，m)：系数r(i，j)；j=1，.，n。  
约束右手边b(I)；i=1，.，m。

第二个数据文件是mknap2。  
这个数据文件包含48个测试问题。  
来自文学作品。这些问题的形式。  
在文件中进行了描述。剩余的数据文件是在P.C.Chu和。J.E.Beasley“多维背包的遗传算法。问题“，”启发式“杂志，第4卷，1998年，第63-86页。这些数据文件是mknapcb1，mknapcb2，.，mknapcb9。

这些数据文件的格式与mknap1的格式相同。这些数据文件每个包含30个测试问题，前十个问题。紧密性比为0.25，后十个问题有紧密性。比率为0.50，最后十个问题的紧密性比率为0.75(请参见。上述文件)。找到的最佳可行解值和LP的值。这些问题的松弛在mkcbres文件中给出。最大的文件是mknapcb9，大小为2000Kb(大约)。整个文件集的大小为5400KB(大约)。

**二、实验环境**

Window10

Microsoft Visual C++

**三、实验原理**

此代码定义了一个以n为列长度，m为行长度的二维数组，应用了二维数组的计算求出符合条件的物品，并求出几种不同搭配的价值数。

首先，此题目中所能放入背包的物品，需要满足10个约束条件才能放入背包中。所以先设置一个循环，该循环是寻找能够符合条件的物品，将循环限制在不超过n的情况，如果循环超过n，则结束该循环。先假设第一个物品是取的，（设置一个数组，放置值0或1，若取该物品则为1,不取该物品则为0。）然后判断它的第一个约束值是否小于右手边约束和，如果符合，则将该物品的约束值放入背包中，如果不符合，则跳出循环。其次，在寻找符合条件的物品时，还需要寻找该物品的10个约束值是否都符合条件，如果符合条件，则取该物品；如果不符合条件，则该物品不取。所以在前面设置寻找符合条件的物品的循环内，再使用一个循环，用于查找该物品的10个约束值是否都符合右手边的约束总值。以上若循环该物品的约束值中发现大于背包剩余的约束总值，则该物品不取（不管接下来的约束值是否符合），如果符合条件，则更新背包价值。通俗地说，就是先假设a物品取，a物品的10个约束条件是否符合，符合则取。在有a物品的前提，看b物品是否符合条件，符合条件则取；…… 假设a不取，b取的条件下，c物品是否符合条件，负责则取；等依次循环。

找出所有符合约束条件的物品放入背包中，计算所得的价值，然后再将所得的几种方案的价值进行对比，并输出，即可得到最大价值的选择方案。

**四、代码**

#include<stdio.h>

#include<math.h>

int a[100];//定义最多100个物品的价值

float b[100][100];//定义数组b[][]表示物品数（变量数）及约束数

float h[100];//定义背包的约束和

int n,m;//定义n为物品数（列长度数），m为约束数（行长度数）

int d;

int s;

int t1,t2;

int cp=0; //当前背包价值

int bestv=0; //迄今最大的价值

int x[100]; //记录在树中的移动路径,为1的时候表示选择该组数据，为0的表示不选择该组数据

void panduan(int i)//定义一个形参int i 用于调用函数

{//printf("%d",i);

if(i==n)//递归结束的判定条件，与下一个else

{

if(cp>bestv)//排序所得的价值，取最大

{

bestv=cp;

printf("%d",bestv);

printf("\n");

}

}

else{

x[i]=1;//假设取

for(int k=0;k<m;k++)//循环取不取的操作次数（行长度数），判断该物品所有约束都符合条件

{

if(h[k]-b[k][i]<0)//如果约束和-所取的约束值0，说明所取约束值不能进入剩下的约束和中去，则不取该物品

{

x[i]=0;//进入值为0的孩子

break;

}

}

if(x[i]==1)//进入该节点的右孩子(值为1的孩子),约束值符合条件取

{

for(int t=0;t<m;t++)//计算约束值循环次数（行长度数）

{

h[t]-=b[t][i];//计算约束值是否符合剩余的约束

}

cp+=a[i];//更新价值

panduan(i+1);

for(int k=0;k<m;k++)//更新约束值得和

{

h[k]+=b[k][i];//更新约束值

}

cp-=a[i]; //此处回溯

}

panduan(i+1);

}

}

void main(){

//输入相关数据变量数组等

printf("请输入物品个数：");

scanf("%d",&n);

printf("请输入约束个数：");

scanf("%d",&m);

printf("请依次输入物品的价值：");

for(s=0;s<n;s++){

scanf("%d",&a[s]);

}

printf("请依次输入物品的约束数:");//输出转化为二维数组输出

for(t1=0;t1<m;t1++)

{

for(t2=0;t2<n;t2++)

{

scanf("%d",&b[t1][t2]);

}

}

printf("请依次输入背包约束和:");

for(d=0;d<m;d++){

scanf("%d",&h[d]);}

panduan(0);}

1. **流程图**

程序开始

假设物品取或者不取

该物品不取

判断该物品的约束条件是否符合右手边约束

循环n个物品

**不符合**

循环判断m个约束条件

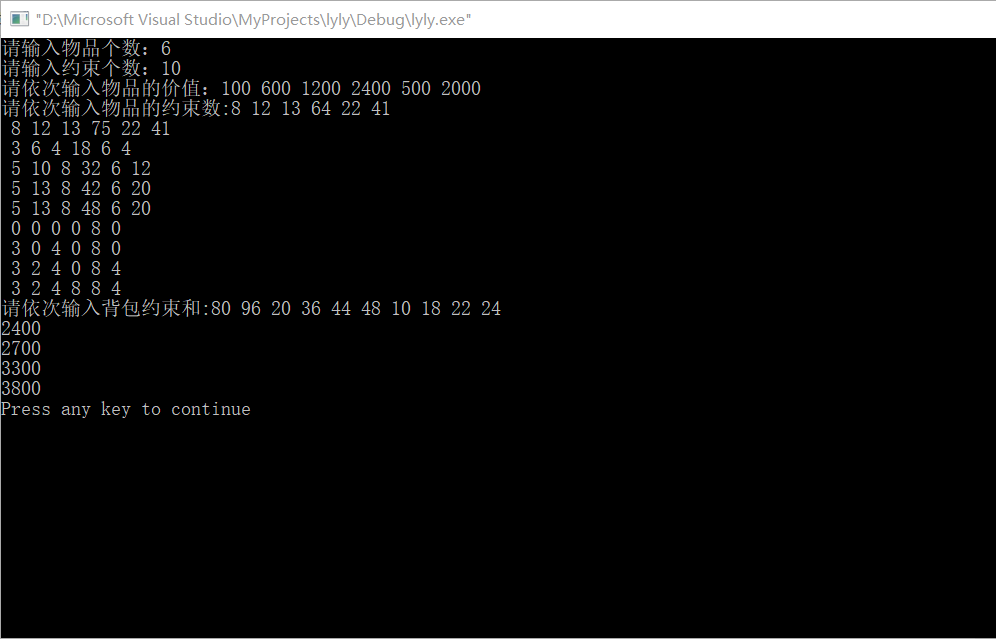
**符合**

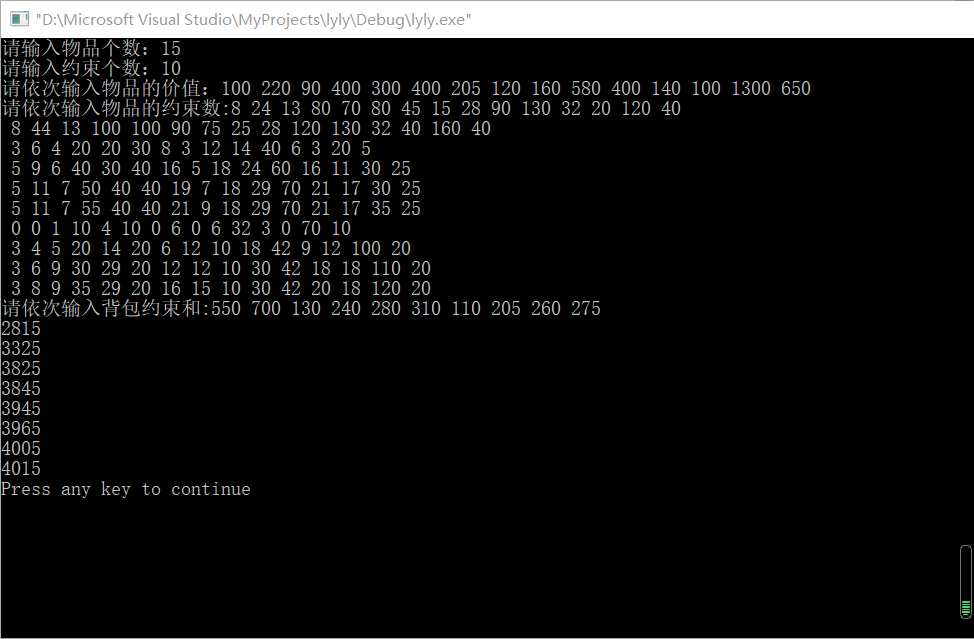
约束值加入相关计算

物品循环结束

程序结束

**六、运行结果（部分）**

****

****