**5-3最小重量：机器设计问题**

问题描述：设某一机器由n个部件组成，每一种部件都可以从m个不同的供应商处购得。设 wij 是从供应商j 处购得的部件i的重量，cij 是相应的价格。试设计一个回溯算法，给出总价格不超过c的最小重量机器设计

算法设计：对于给定的机器部件重量和机器部件价格，计算总价格不超过c的最小重量机器设计。回溯法要求要给出约束条件，很明显：总价格不超过c，设当前已选部件的重量和为cw，价格之和为cc，当前最优重量用bestw表示，初始化bestw=10000；限界条件：cw<bestw。

3.程序代码：

#include<iostream>

#include<stdio.h>

using namespace std;

int w[100][100];//w[i][j]为第i个零件在第j个供应商的重量

int c[100][100];//c[i][j]为第i个零件在第j个供应商的价格

int bestx[100];//bestx[i]表示一次搜索到底后的最优解，用来存放第i个零件的供应商，

int x[100];//x[i]临时存放第i个零件的供应商

int cw=0,cc=0,bestw=10000;

int cost//限定价格

int n;//部件数

int m;//供应商数

void Backtrack(int t)

{

int j;

if(t>n)//搜索到叶子结点，一个搜索结束，所有零件已经找完

{

bestw=cw;//当前最小重量

for(j=1;j<=n;j++)

bestx[j]=x[j];

}

else

{

for(j=1;j<=m;j++)

{

if(cc+c[t][j]<=cost && cw+w[t][j]<bestw)

{

x[t]=j;

cc+=c[t][j];

cw+=w[t][j];

Backtrack(t+1);

cc-=c[t][j];

cw-=w[t][j];

}

}

}

}

int main()

{

int i,j;

cout<<"请输入部件数："<<endl;

cin>>n;

cout<<"请输入供应商数："<<endl;

cin>>m;

cout<<"请输入限定价格："<<endl;

cin>>cost;

cout<<"请输入各部件的在不同供应商的重量："<<endl;

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

for(j=1; j<=m; j++)

cin>>w[i][j];

cout<<"请输入各部件的在不同供应商的价格："<<endl;

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

for(j=1; j<=m; j++)

cin>>c[i][j];

Backtrack(1);

cout<<"每个部件的供应商："<<endl;

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

cout<<bestx[i]<<' ';

cout<<endl;

cout<<bestw;

return 0;

}

/\*

测试数据:

3

3

7

1 2 3

3 2 1

2 3 2

1 2 3

5 4 2

2 1 2

\*/