5-3（I）

算法分析：

int n;

int m;

int w[100][100];

int c[100][100];

int bestx[100];//最优解

int bestw=1<<18;//当前最小重量

int x[100];//当前解

int cw=0;

int cc=0;

int d;

本题与装载问题类似，可以用回溯法解决。如上，当节点到达叶子节点时，如果目前解小于最优解，那么最优解就是被目前解替换，否则，当目前重量大于最优解的重量，或者目前价格大于d，那么就剪掉，否则继续下一层检索。

代码：

#include<iostream>

using namespace std;

int n;

int m;

int w[100][100];

int c[100][100];

int bestx[100];//最优解

int bestw=1<<18;//当前最小重量

int x[100];//当前解

int cw=0;

int cc=0;

int d;

int main()

{

void backtrack(int i);

cin>>n>>m>>d;

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

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

cin>>c[i][j];

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

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

cin>>w[i][j];

backtrack(1);

cout<<bestw<<endl;

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

cout<<bestx[i]<<" ";

return 0;

}

void backtrack(int i)

{

if(i>n)

{ if(cw<bestw||x[i]==-1){

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

bestx[j]=x[j];

bestw=cw;

return;

}

}

else

{

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

{

cw+=w[i][j];

cc+=c[i][j];

x[i]=j;

if(cc<=d&&cw<bestw)

backtrack(i+1);

cw-=w[i][j];

cc-=c[i][j];

x[i]=-1;

}

}

}

