**蛮力法解决0-1背包问题**

【代码】

#include <iostream>

using namespace std;

#define N 25

int max\_value = 0;

int result[100];

int max\_cur = 0;

using namespace std;

struct good

{

int weight;

int value;

int flag;//记录该货物是否被挑选

}goods[100];

void print\_subset(int n, int \*A, int cur,int maxWeight)

{

int sum = 0,count = 0;

cout << "{";

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

{

printf("%d ", A[i] + 1);//输出子集 当前的集合

sum += goods[A[i]].value;

count += goods[A[i]].weight;

}

cout << "}";

for (int i = 0; i < n - cur; i++) //用于调整格式

cout << " ";

cout << "\t" << count;

cout << "\t" << sum;

if ((max\_value < sum) && (count <= maxWeight)) //打擂台获取最优值

{

max\_value = sum;

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

result[i] = A[i];

max\_cur = cur;

}

if (count > maxWeight)

cout << "\t" << " " << "超重" << endl;

else

cout << endl;

int s = cur ? A[cur - 1] + 1 : 0;//确定当前最小的可能值 如果这里不是 这里特殊的就是cur==0时 其他的就是选比前一个大1的

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

{

A[cur] = i;//将i加入当前的集合

print\_subset(n, A, cur + 1,maxWeight);// 递归构造子集

}

}

int main()

{

int A[100];

int goodNum = 0, maxWeight = 0;

cout << "请输入物品的个数：";

cin >> goodNum;

cout << "请输入背包的总容量：";

cin >> maxWeight;

cout << "请输入对应物品的重量和价值：" << endl;

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

{

goods[i].flag = 0;

cin >> goods[i].weight >> goods[i].value;

}

cout << "装入物品子集" << " " << "总重量" << " " << "总价值" << "\t" << "选择情况" << endl;

print\_subset(goodNum, A, 0,maxWeight);

cout << endl << "最终结果：" << endl << "{";

for (int i = 0; i < max\_cur; i++)

printf("%d ", result[i] + 1);//输出子集 当前的集合

cout << "}";

cout << "最大价值为：";

for (int i = 0; i < max\_cur - 1; i++)

cout << goods[result[i]].value << "+";

cout << goods[result[max\_cur - 1]].value << " = ";

cout << max\_value << endl;

system("pause");

return 0;

}

【效果】

