**动态规划法解决0-1背包问题**

【代码】

#include<iostream>

#include<graphics.h>

#include <conio.h>

using namespace std;

#define M 100

void Knapsack(int v[], int w[], int c, int n, int m[M][101])

{

char s[5];

int jmax = min(w[n - 1] - 1, c); //取当中的最小值

for (int j = 0; j <= jmax; j++) //当第n个物品不选时，m[n][j]价值为0

m[n][j] = 0;

for (int j = w[n - 1]; j <= c; j++) //当第n个物品选择时，m[n][j]价值为value[n]

m[n][j] = v[n - 1];

//自n-1到2逐层计算各m[i][j] 的值

for (int i = n - 1; i >= 1; i--)

{

jmax = min(w[i - 1] - 1, c);

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

m[i][j] = m[i + 1][j];

for (int j = w[i - 1]; j <= c; j++)

m[i][j] = max(m[i + 1][j], m[i + 1][j - w[i - 1]] + v[i - 1]);

}

m[1][c] = m[2][c];

if (c >= w[0]) //处理第一层的边界条件

m[1][c] = max(m[1][c], m[2][c - w[0]] + v[0]);

cout << endl << "构成的m数组为:" << endl;

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

{

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

printf("%3d ", m[i][j]);

printf("\n");

}

printf("\n");

setlinestyle(PS\_SOLID,2); //实线2像素的线

//画整个矩形

for (int m = 20; m <= (c + 1) \* 40 + 20; m += 40)

line(m,10,m,c \* 20 + 10);

for (int k = 10; k <= n \* 40 + 10; k += 40)

line(20, k, (c + 1) \* 40 + 20, k);

//将m数组的书存入方格当中

for (int i = (n - 1) \* 40 + 25,p = n; i >= 25,p >= 1; i -= 40,p --)

{

for(int j = 35,n = 0;j < (c + 1) \* 40,n <= c;j += 40,n ++)

{

sprintf\_s(s,"%d", m[p][n]);

outtextxy(j,i, s);

Sleep(100);

}

}

}

//去除满足条件的

void Traceback(int m[M][101], int w[], int c, int n, int x[])

{

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

if (m[i][c] == m[i + 1][c])

x[i] = 0;

else

{

x[i] = 1;

c = c - w[i - 1];

}

x[n] = (m[n][c]) ? 1 : 0;

}

int main()

{

int number, capacity, weight[M], value[M];

int m[M][101];

int x[M];

FILE \*fp = NULL;

fp = fopen("D:\\Desktop\\test.txt", "a+");

if (fp == NULL) {

cout << "读取文件失败！" << endl;

}

fscanf\_s(fp, "%d", &number);

fscanf\_s(fp, "%d", &capacity);

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

fscanf\_s(fp,"%d",&weight[i]);

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

fscanf\_s(fp, "%d", &value[i]);

cout << "读取的物品个数：" << number << endl;

cout << "读取的背包的总容量：" << capacity << endl;

cout << "各个物品的重量跟价值：" << endl;

cout << "编号" << "\t" << "物品重量" << "\t" << "物品价值" << endl;

for (int k = 0; k < number; k++)

cout << k << "\t " << weight[k] << "\t\t " << value[k] << endl;

initgraph(640, 480); //创建窗口

Knapsack(value, weight, capacity, number, m);

Traceback(m, weight, capacity, number, x);

\_getch(); // 按任意键继续

closegraph(); // 关闭图形界面

cout << "物品编号\t\t物品重量\t\t物品价值\t\t该物品是否放入背包(1表示放入)" << endl;

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

printf("%4d\t\t%12d\t\t%12d\t\t%16d\n", i + 1, weight[i], value[i], x[i + 1]);

cout << endl;

cout << "整个背包的价值总和为:" << m[1][capacity] << endl;

system("pause");

return 0;

}

【截图】

