**贪心算法解决背包问题**

【代码】

#include<iostream>

#include<graphics.h>

#include <conio.h>

using namespace std;

#define N 25

struct good {

int weight;

int value;

}goods[N];

void graphic(int goodsNum,float x[])

{

setlinecolor(WHITE);

line(0, 540, 640, 540);

line(640, 540, 630, 530);

line(640, 540, 635, 550);

line(20, 200, 20, 540);

line(20, 200, 10, 210);

line(20, 200, 30, 210);

char s[40] = "价值 / 重量 （v[ i ] / w[ i ]）";

outtextxy(30, 200, s);

settextcolor(WHITE);

char s1[17] = "重量（w[ i ]）";

outtextxy(620, 550, s1);

settextcolor(WHITE);

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

{

if (x[i] >= 1)

{

setfillcolor(LIGHTGREEN);

fillrectangle(goods[i].weight \* 10, 540 - ((goods[i].value / goods[i].weight) \* 30), goods[i].weight \* 10 + 40, 540);

}

else if(x[i] > 0 && x[i] < 1) {

setfillcolor(LIGHTGRAY);

fillrectangle(goods[i].weight \* 10, 540 -

((goods[i].value / goods[i].weight) \* 30),

goods[i].weight \* 10 + 40,

540 - ((goods[i].value / goods[i].weight) \* 30 \* x[i]));

setfillcolor(LIGHTGREEN);

fillrectangle(goods[i].weight \* 10, 540 -

((goods[i].value / goods[i].weight) \* 30 \* x[i]),

goods[i].weight \* 10 + 40, 540);

}

else {

setfillcolor(LIGHTGRAY);

fillrectangle(goods[i].weight \* 10, 540 -

((goods[i].value / goods[i].weight) \* 30),

goods[i].weight \* 10 + 40, 540);

}

char str[15];

sprintf\_s(str, "%.1f", x[i]);

outtextxy(goods[i].weight \* 10 + 15, 540 - ((goods[i].value / goods[i].weight) \* 30 / 2), str);

char str1[5];

sprintf\_s(str1, "%d", goods[i].weight);

int j = goods[i].weight \* 10 + 10;

outtextxy(j, 550, str1);

char str2[5];

sprintf\_s(str2, "%d", goods[i].value / goods[i].weight);

outtextxy(j + 5, 520 - ((goods[i].value / goods[i].weight) \* 30), str2);

}

}

void quick\_sort(int low, int high)

{

int i, j;

if (low < high) {

i = low;

j = high;

goods[0] = goods[i];

do {

while ((goods[j].value / goods[j].weight < goods[0].value / goods[0].weight) && i < j)

j--;

if (i < j)

{

goods[i] = goods[j];

i++;

}

while ((goods[i].value / goods[i].weight > goods[0].value / goods[0].weight) && i < j)

i++;

if (i < j)

{

goods[j] = goods[i];

j--;

}

} while (i != j);

goods[i] = goods[0];

quick\_sort(low, i - 1);

quick\_sort(i + 1, high);

}

}

float Knapsack(float x[N], float C, int n)

{

int i;

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

x[i] = 0;

i = 1;

float total = 0;

while (goods[i].weight < C)

{

x[i] = 1;

total = total + goods[i].value;

C = C - goods[i].weight;

i++;

}

x[i] = C / goods[i].weight;

total = total + x[i] \* goods[i].value;

return total;

}

int main()

{

float weight[N], value[N];

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 = 1; i <= goodNum; i++)

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

float x[N];

initgraph(840, 640);

quick\_sort(1, goodNum); //按照value / weight 从大到小排列

float result = Knapsack(x, maxWeight, goodNum);

graphic(goodNum, x);

\_getch();

closegraph();

cout << "物品" << "\t" << "重量（w）" << "\t" << "价值（v）" << "\t" << "价值/重量（v/ w）" << "\t" << "选中比例" << endl;

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

cout << i << "\t" << goods[i].weight << "\t\t " << goods[i].value << "\t\t\t" << goods[i].value / goods[i].weight << "\t\t" << x[i] << endl;

cout << "背包总重量：" << result << endl;

system("pause");

return 0;

}

【演示】



