西安交通大学实验报告

课程名称： 算法设计与问题求解 实验名称： 贪心算法-穷举法

学 院： 机械工程学院 实验日期 2020 年 11 月7 日

班 级： 机械97班 姓 名： 杨逢诜 学号： 2193712613

#### 一、实验内容和结果

* **题目1 ：使用贪心算法求解0-1背包问题**

0-1背包问题

给定n种物品和一个背包。物品i的重量是Wi，其价值为Vi，背包的容量为W。问应如何选择物品装入背包，使得装入背包中的物品的总价值最大?在选择物品装入背包时，对每种物品i只有两种选择，要么装入，要么不装入，不能将物品j装入背包多次，也不能只装入物品 j 的一部分。

假设有三种物品，分别为1，2，3，其重量分别为20，30，60，（kg），对应的价值分别是40，90，240，背包容量为90kg。

物品1每公斤价值40/20=2

物品2每公斤价值是90/30=3

物品3每公斤价值是240/60=4

要使装入的物品总价值最大，则需要先装大价值的物品，最后装最小价值的。所以先装物品3，把60公斤均装进背包，还剩30公斤，把物品2均装入30公斤，此时包的容量已经装满，此时的背包中的物品的总价值为240+90=330为最大总价值。

算法：首先计算每种物品单位重量的价值vi／wi，然后，依贪心选择策略，将尽可能多的单位重量价值高的物品装入背包。算法的主要计算时间在于将各种物品依其单位重量的价值从大到小排序。因此，算法的计算时间上界为O(nlogn)。

编程实现上述方案.

【源程序】

#include<stdio.h>

int MAX=90; //定义背包容量

struct iteminfo //定义结构体：物品，包含其编号、质量、价值、价值比

{

int order;

int mass;

int value;

double vpm;

};

struct packageinfo //定义结构体：背包，包含其总质量和其中物体的编号

{

int mass;

int item[20];

};

int main() //主函数开始；

{

int n,i,j; //声明变量：物品组、背包、物品总数、循环控制变量

struct iteminfo items[20];

struct packageinfo package;

package.mass=0; //初始化背包质量

printf("请输入物品个数:\n"); //输出提示，将物品个数填入存储空间中

scanf("%d",&n);

for(i=0;i<n;i++) //初始化背包空间：里面什么物品都没有（都标记为0）

{

package.item[i]=0;

}

printf("请输入各物品的价值:\n"); //输出提示，将物品价值、物品质量填入存储物品信息的结构体数组中，为物品编号

for(i=0;i<n;i++)

{

scanf("%d",&items[i].value);

}

printf("请输入各物品的重量:\n");

for(i=0;i<n;i++)

{

scanf("%d",&items[i].mass);

items[i].vpm=(double)items[i].value/(double)items[i].mass;

items[i].order=i;

}

for(i=0;i<n-1;i++) //根据价值比为物品排序，从高到低

{

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

{

if(items[i].vpm<items[j].vpm)

{

struct iteminfo item=items[i];

items[i]=items[j];

items[j]=item;

}

}

}

for(i=0;i<n;i++) //从价值最高的物品开始，将其逐个填入背包直至填满

{

if(package.mass+items[i].mass<=MAX)

{

package.mass=package.mass+items[i].mass;

package.item[items[i].order]=1; //将背包结构体中该物品在数组中对应的位置填入1，表示该物品已经放入

}

else break; //若背包已填满则不再执行填充

}

printf("物品建议取法:(0--不装，1--装)\n");

for(i=0;i<n;i++) //根据题设格式输出结论，主函数结束

{

printf("x[%d] ",i);

}

printf("\n");

for(i=0;i<n;i++)

{

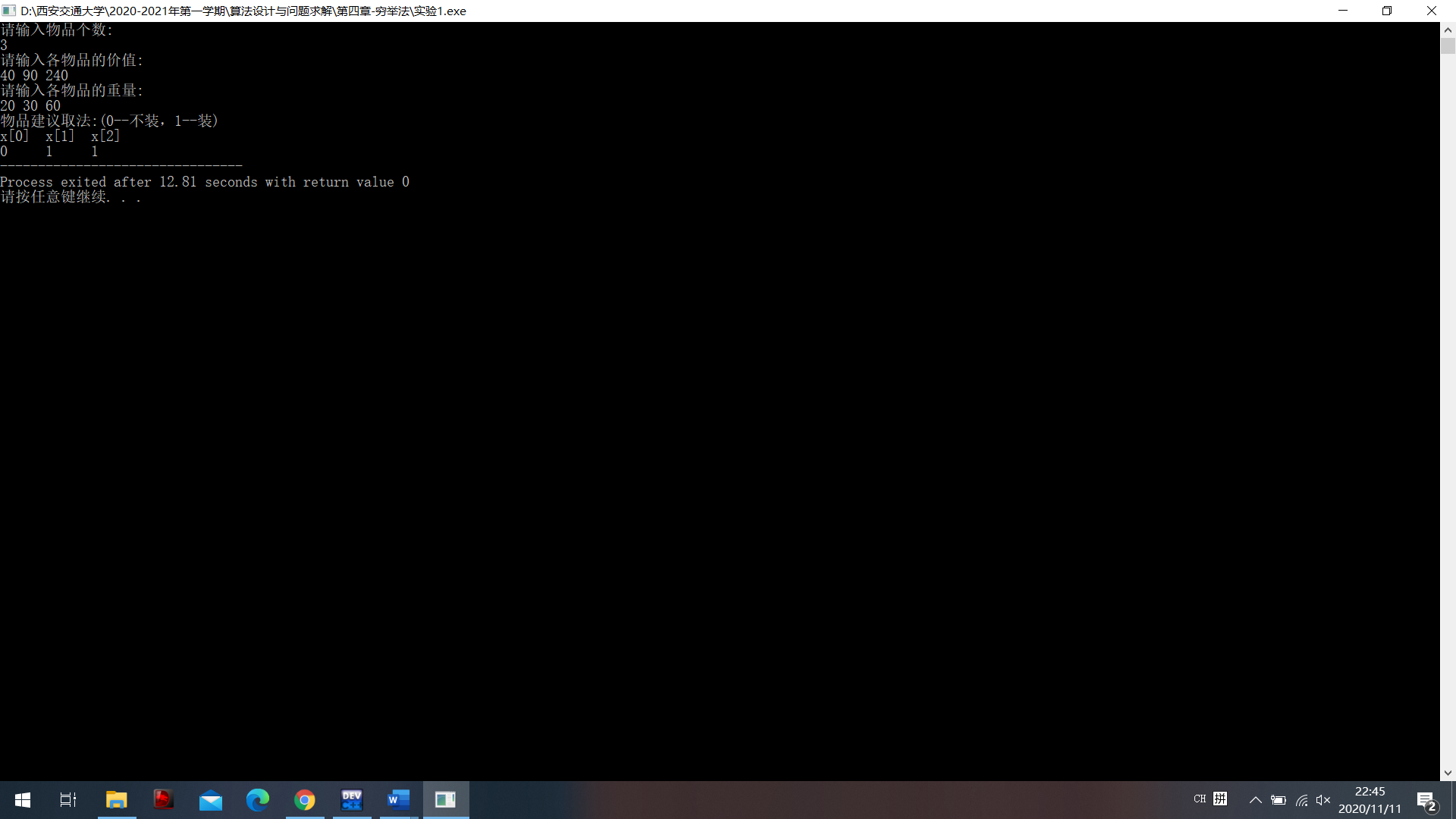
printf("%d ",package.item[i]);

}

return 0;

}

【运行结果】



* **题目2：使用穷举法求解0-1背包问题**

算法：

设输入物品为 w1, w2,..., wn

v1, v2,....,vn

设长度为n的2进制数（不足n位的前面补0到n位）的第i位为1表示取第i件，为0表示不取第i件物品. 则所有这样的01组合代表一种取法。这样的2进制数的范围是0～2n-1.

于是，作循环令i从0～2n-1，将i逐位拆开放到长度为 n整型数组X中，计算这种取法对应的重量和价值（Σw[i]\*x[i]等）。计算问题最优解。

编程实现上述方案。

【源程序】

#include<stdio.h>

int MAX=90; //定义背包容量

struct iteminfo //定义结构体：物品，包含质量、价值

{

int mass;

int value;

};

struct packageinfo //定义结构体：背包，包含其总质量、总价值和其中物体的编号

{

int mass;

int value;

int items[20];

};

struct packageinfo package; //声明全局变量：背包，以防某些函数内运算结果不能带出

int item[20]; //声明全局变量：物品组，用于指征背包内可能填充的特定物品组合

void Selection(struct iteminfo \*items,int k,int n)//声明遍历函数，n为尚未填入背包的物品数

{

int i,j; //声明循环控制变量

if(k==1) //若只剩下一个空没有填入，填入那个空之后直接开始判定：

{

for(i=0;i<2;i++) //遍历两种情形，填入或者不填，给对应数组赋值，0是不填，1是填

{

item[k-1]=i;

int mass=0,value=0; //声明变量：物品组的总质量、总价值

for(j=0;j<n;j++)

{

mass=mass+item[j]\*items[j].mass;

value=value+item[j]\*items[j].value;

}

//for(j=0;j<n;j++)

//{

//printf("%d ",item[j]);

//}

if(mass<=MAX&&value>=package.value)//若物品组总质量小于背包可容的最大质量，且物品组的价值小于当前已经取得的物品组价值最大值

{

package.value=value; //将该物品组的信息填入背包结构体中，先填入质量和价值，再利用循环填入物品组

package.mass=mass;

//printf("mass:%d value:%d ",package.mass,package.value);

for(j=0;j<n;j++)

{

package.items[j]=item[j];

}

}

//for(j=0;j<n;j++)

//{

//printf("%d ",package.items[j]);

//}

//printf("\n");

}

}

else //若还剩下多个物品未确定，则先确定下一个物品的两种可能，然后接着调用函数自身计算下下个，以此类推直到最后一层

{

for(i=0;i<2;i++)

{

item[k-1]=i;

Selection(items,k-1,n);

}

}

}

int main() //主函数开始；

{

int n,i; //声明变量：物品组、物品总数、循环控制变量

struct iteminfo items[20];

printf("请输入物品个数:\n"); //输出提示，将物品个数填入存储空间中

scanf("%d",&n);

for(i=0;i<n;i++) //初始化背包空间：里面什么物品都没有（都标记为0）

{

item[i]=0;

}

printf("请输入各物品的价值:\n"); //输出提示，将物品价值、物品质量填入存储物品信息的结构体数组中

for(i=0;i<n;i++)

{

scanf("%d",&items[i].value);

}

printf("请输入各物品的重量:\n");

for(i=0;i<n;i++)

{

scanf("%d",&items[i].mass);

}

package.value=0; //初始化背包空间：没有任何有价值物品

Selection(items,n,n); //调用函数算出所有可能值中最好的结论；

printf("物品建议取法:(0--不装，1--装)\n");

for(i=0;i<n;i++) //根据题设格式输出结论，主函数结束

{

printf("x[%d] ",i);

}

printf("\n");

for(i=0;i<n;i++)

{

printf("%d ",package.items[i]);

}

return 0;

}

【运行结果】

