

数据结构

2022

# 实验报告

实验项目名称：递归

班级：2021级6班

学号：2021302181138

姓名：伍旺旺

指导教师：沈志东

实验时间：2022年4月21日

## 实验五: 递归

### 一、实验要求

实验一、采用递归和非递归方法求解 Hanoi 问题

实验二、用递归方法求解 0/1 背包问题

### 二、实验环境

硬件：微型计算机

软件：Windows 操作系统、Microsoft Visual Studio Code

### 三、实验步骤及思路

实验一、采用递归和非递归方法求解 Hanoi 问题

- 题目分析

本实验的功能算法如下：

```
Hanoi1(int n,char a,char b,char c): 求解 Hanoi 问题的递归算法  
Hanoi2(int n,char x,char y,char z): 求解 Hanoi 问题的非递归算法
```

栈的基本运算算法如下：

```
void InitStack(StackType *&s);
void DestroyStack(StackType *&s);
bool StackEmpty(StackType *s);
bool Push(StackType *&s, ElemType e);
bool Pop(StackType *&s, ElemType &e);
```

- 实验具体步骤

## 1. 递归算法编写:

```
void Hanoi1(int n, char a, char b, char c)
{
    if(n == 1)
        printf("\t将第 %d 个盘片从 %c 移动到 %c\n", n, a, c);
    else
    {
        Hanoi1(n-1, a, c, b);
        printf("\t将第 %d 个盘片从 %c 移动到 %c\n", n, a, c);
        Hanoi1(n-1, b, a, c);
    }
}
```

## 2. 非递归算法编写:

```

void Hanoi2(int n,char x,char y,char z)
{
    StackType *st;
    ElemType e,e1,e2,e3;

    if(n <= 0)
        return;
    InitStack(st);

    e.n = n;
    e.x = x;
    e.y = y;
    e.z = z;
    e.flag = false;

    Push(st,e);
    while(!StackEmpty(st))
    {
        Pop(st,e);
        if(e.flag == false)
        {
            e1.n = e.n - 1;
            e1.x = e.y;
            e1.y = e.x;
            e1.z = e.z;
            if(e1.n == 1)
                e1.flag = true;
            else
                e1.flag = false;
            Push(st,e1);
            e2.n = e.n;
            e2.x = e.x;
            e2.y = e.y;
            e2.z = e.z;
            e2.flag = true;
            Push(st,e2);
            e3.n = e.n - 1;
            e3.x = e.x;
            e3.y = e.z;
            e3.z = e.y;
            if(e3.n == 1)
                e3.flag = true;
            else e3.flag = false;
            Push(st,e3);
        }
        else
            printf("\t将第 %d 个盘片从 %c 移动到 %c\n",e.n,e.x,e.z);
    }
    DestroyStack(st);
}

```

## 实验二、用递归方法求解 0/1 背包问题

- 题目分析

功能算法如下：

```
void knap(int i,int tw,int tv,int op[]): 求解 0/1 背包问题  
void DispASolution(int x[],int n): 输出 x 中保存的一个解
```

定义以下全局变量：

```
int maxv;           //存放最优解的总价值  
int maxw;           //存放最优解的总重量  
int x[MAXN];        //存放最终解  
int W = 7;          //限制的总重量  
int n = 5;          //物品总数  
int w[] = {5,3,2,1,1}; //物品重量  
int v[] = {5,4,4,3,1}; //物品价值
```

- 实验具体步骤

1. 编写knap()函数:

```

void knap(int i,int tw,int tv,int op[])
{
    int j;

    if(i >= n)
    {
        if(tw <=W && tv > maxv)
        {
            maxv = tv;
            maxw = tw;
            for(j = 1;j <= n;j++)
                x[j] = op[j];
        }
    }
    else
    {
        op[i] = 1;
        knap(i + 1,tw + w[i],tv + v[i],op);
        op[i] = 0;
        knap(i + 1,tw,tv,op);
    }
}

```

## 2. 编写DispASolution()函数:

```

void DispASolution(int x[],int n)
{
    int i;


    printf("最佳装填方案是: \n");
    for(i = 1;i <= n;i++)
        if(x[i] == 1)
            printf("\t选取第 %d 个物品\n",i);
    printf("\n总重量 = %d, 总价值 = %d\n",maxw,maxv);
}

```

# 四、实验结果及分析

## 实验一、采用递归和非递归方法求解 Hanoi 问题

实验结果如下：

 选择 C:\Windows\system32\cmd.exe

递归算法：3 个盘片移动过程：

```
将第 1 个盘片从 X 移动到 Z
将第 2 个盘片从 X 移动到 Y
将第 1 个盘片从 Z 移动到 Y
将第 3 个盘片从 X 移动到 Z
将第 1 个盘片从 Y 移动到 X
将第 2 个盘片从 Y 移动到 Z
将第 1 个盘片从 X 移动到 Z
```


非递归算法：3 个盘片移动过程：

```
将第 1 个盘片从 X 移动到 Z
将第 2 个盘片从 X 移动到 Y
将第 1 个盘片从 Z 移动到 Y
将第 3 个盘片从 X 移动到 Z
将第 1 个盘片从 Y 移动到 X
将第 2 个盘片从 Y 移动到 Z
将第 1 个盘片从 X 移动到 Z
```

请按任意键继续. . .

## 实验二、用递归方法求解 0/1 背包问题

实验结果如下：

 选择 C:\Windows\system32\cmd.exe

最佳装填方案是：

```
选取第 1 个物品
选取第 2 个物品
选取第 3 个物品
选取第 4 个物品
```

总重量 = 7，总价值 = 12

请按任意键继续. . . ■

此处物品总数为 5，其重量分别是 5，3，2，1，1，价值分别是 5，4，4，3，1，限制的总重量为 7。最佳方案是选择 4 物品，总重量为 7，总价值为 12。

## 五、总结

- 通过采用递归和非递归方法求解 Hanoi 问题，领会了基本递归算法设计和递归到非递归的转换方法。
- 通过用递归方法求解 0/1 问题更进一步掌握了递归的算法设计。