

# lab1

宋昕

May 15, 2019

## Contents

<b>1 lab1 分治法实验</b>	<b>1</b>
1.1 任务一	1
1.2 任务二 (计算机算法设计与分析, 第五版) P41 开始, 算法实现题目	1
1.2.1 2-1 众数问题 (要求用分治法, 即递归的方法完成)	1
1.2.2 2-3 半数集问题	2
1.2.3 2-5 有重复元素的排列问题	2
1.2.4 2-6 字典序问题	2
1.2.5 2-7 集合划分问题 (1)	3
1.2.6 2-8 集合划分问题 (2)	3
1.2.7 2-9 双色汉诺塔问题	3
1.2.8 2-11 整数因子分解问题	3

## 1 lab1 分治法实验

### 1.1 任务一

编写 BubbleSort.c 和 MergeSort.c, 利用提供的数据集, 分别记录不同大小数组的排序时间, 并根据规模和时间画出两个算法的实际测量时间复杂度。

### 1.2 任务二 (计算机算法设计与分析, 第五版) P41 开始, 算法实现题目

#### 1.2.1 2-1 众数问题 (要求用分治法, 即递归的方法完成)

第一步: 对集合 (数组) 中的元素进行排序  
第二步: 设计一个递归函数 Mode, 输入参数是排序过的一维数组以及数组的起点和终点, 返回值是一维数组的众数  
第三步: 设计 main 函数, 并测试

```
int Mode(int a[], int i, int j) {
    if (i == j) return 1;

    int m = (i + j) / 2;

}
```

### 1.2.2 2-3 半数集问题

第一步：理解问题，用小规模输入进行模拟演算，确保理解题意第二步：进行递归式思考，分析问题的输入和输出，原问题的输入/出是什么，子问题的输入/出是什么，以及原问题分解为几个子问题，基本情况的条件和值是什么？第三步：尝试写出计算半数集的递推式，递推式中的多个子问题会用到求和公式  $\Sigma$ ，要明确求和的上、下边界第四步：把递推公式改写为程序代码 HalfSet.c

```
cat HalfSet.c
```

### 1.2.3 2-5 有重复元素的排列问题

第一步：与无重复元素的排列问题作比较，找到重复发生的环节，然后设置去重的条件第二步：编码实现

```
cat PermRemove.c
```

### 1.2.4 2-6 字典序问题

第一步：观察无重复全排列的次序（用课堂上的讲解的算法）和字典序次序的差异，进而发现字典序和的差异在于从序列中选出一个打头元素后，当前打头元素该放在什么位置？课堂算法的做法是当前打头元素和选出的新打头元素交换位置，而字典序是把选出的打头元素放在当前打头元素之前，选出打头元素后，其空下的位置由后续元素向前补充。因此需要修改全排列生成算法中的 Swap 功能。第二步：编码实现第一步的构思

```
cat DictOrder.c
```

第三步：分析推算 2, 6, 4, 5, 8, 1, 7, 3 序列的次序（应为 8228），从推算过程中看看能否写成一个递归式：

$$f(i, n, seq) = f(i + 1, n, seq) + (order(A[i], seq) - 1) * (n - 1)!$$

, if

$$i + 1 = n$$

,then

$$f(i, n, seq) = 1$$

#### **1.2.5 2-7 集合划分问题 (1)**

#### **1.2.6 2-8 集合划分问题 (2)**

#### **1.2.7 2-9 双色汉诺塔问题**

双色汉诺塔问题和单色汉诺塔问题在输入规模一样的情况下，挪动盘子的次数是一样的。但过程却是不同的，单色汉诺塔有两个子问题，双色汉诺塔有四个子问题，双色公式

#### **1.2.8 2-11 整数因子分解问题**