

# 南京邮电大学 2021/2022 学年第二学期

## 《算法分析与设计》期末试卷（A）

院(系)\_\_\_\_\_ 班级\_\_\_\_\_ 学号\_\_\_\_\_ 姓名\_\_\_\_\_

题号	一	二	三	总分
得分				

得分

### 一、证明题（本大题共 2 小题，每小题 10 分，共 20 分）

1. 对下面搜索算法的正确性进行证明。并举例说明。

```
int Search(int *l, int n, int x) const
{
    int m, left=0, right=n-1;
    while(left<=right)
    {
        m=(left+right)/2;
        if(x<l[m]) right=m-1;
        else if (x>l[m]) left=m+2;
        else return m;
    }
    return -1;
}
```

2. 已知  $f_1(n)=O(g_1(n))$ ,  $f_2(n)=O(g_2(n))$ , 证明  $f_1(n)+f_2(n)=O(g_1(n)+g_2(n))$ 。

得分

### 二、算法题（本大题共 5 小题，共 60 分）

1. 迭代两路合并排序算法思想是：将有  $n$  个元素的序列看成是  $n$  个长度为 1 的有序子序列，然后两两合并长度为 1 的子序列，得到  $\lfloor n/2 \rfloor$  个长度为 2 或 1 的有序子序列；再两两合并长度为 2 的子序列……直到有序。（10 分）

- (1) 结合上述算法说明它采用的是哪种算法设计策略。为了实现该算法，需要额外设计什么数据结构。
- (2) 对关键字序列（30, 19, 17, 33, 20, 9, 56, 48）进行迭代两路合并排序，给出各趟排序结果和最坏情况下的时间复杂度。

2. 子集和数问题的回溯法算法 SumOfSub 的前置条件是  $n$  个正数  $w_i$  已按非减次序排列。现有两组数据  $W_1=(1,2,3,4)$  和  $W_2=(4,3,2,1)$ ，正整数  $M=5$ 。若对这两组数据执行 SumOfSub 算法（不对  $W_2$  排序），试分析它们在执行时间上的差异。（10 分）

3. 设有两个序列 $(x_1, \dots, x_5) = (a, c, b, c, a)$ ,  $(y_1, \dots, y_4) = (a, b, a, c)$ 。(15 分)
- (1) 计算最长公共子序列长度的二维数组  $c$  的值, 并给出最长公共子序列的长度。
  - (2) 在上述计算出的  $c$  数组上, 求最长公共子序列。需用箭头画出回溯的路径。

4. 下图是 8-谜问题的初始状态和目标状态。(15 分)

1	2	3
4		5
7	8	6

初始状态

1	2	3
4	5	6
7	8	

目标状态

- (1) 判断该初始状态能否到达目标状态 (需给出计算过程)。
  - (2) 设计用 LC 分枝限界法求解 8 谜问题的算法步骤。
  - (3) 用 LC 分枝限界法, 从上述初始状态出发, 画出实际生成的状态空间树。要求在结点旁标上  $\hat{c}(X)$  的值。
5.  $n$ -皇后问题要求在一个  $n \times n$  的棋盘上放置  $n$  个皇后, 使得它们彼此不受“攻击”。由于  $n$ -皇后问题的解具有对称性, 因此可以不必搜索状态空间树中的全部结点, 达到提高算法效率之目的。具体做法是, 当输出一个解时, 即利用对称性输出另一个解。请编程实现该算法(Place 函数可直接调用, 不用编程)。(10 分)。

得分

### 三、应用题 (本大题共 2 小题, 每小题 10 分, 共 20 分)

1. RSA 是第一个较完善的公开密钥算法, 其中的加密公式为  $C = M^e \bmod n$ 。  
请编程实现对明文  $M$  加密的函数 encrypt。已知  $e=5$ ,  $n=35$ ,  $M=12$ , 计算密文  $C$  (含过程)。(10 分)

2. 会场安排问题描述如下:

设有  $n$  个活动的集合  $E = \{1, 2, \dots, n\}$ , 其中每个活动都要求占有同一资源(会场), 而在同一时间内只能有一个活动占有该资源。每个活动给出其占有该资源的起始时间  $s_i$  和结束时间  $f_i$ , 且  $s_i < f_i$ 。若选择了活动  $i$ , 则它在半开半闭区间  $[s_i, f_i)$  内占用资源。若区间  $[s_i, f_i)$  与  $[s_j, f_j)$  不相交, 即  $s_i \geq f_j$  或  $s_j \geq f_i$ , 则称活动  $i$  与  $j$  是相容的。会场安排问题就是要在所给的活动集合中选出最大的相容活动子集合使得可以安排的活动数最多, 即从整体上看, 就是使会场 (子集合) 的数量最少。

运用所学知识求解会场安排问题。(10 分)

- (1) 说明该问题可用哪种算法设计方法 (策略) 求解, 并说明理由, 给出求解该问题的策略。
- (2) 给出会场安排问题算法的前置条件。
- (3) 请补充完成下面的会场安排问题算法:

```
int Algo(int *s, int *f, int n)
{
    int i, j;
    int sum; //需要的会场个数
    ①: sum=0;
```

```
for(i=0; i<n; i++)
{   if(____②____) sum++; //另开辟一个会场，取下一个活动 i
    else j++;           //取前面会场的最后一个活动 j
}
____③____;
}
```