

## 算法基础

第二次作业 (DDL: 2024 年 9 月 25 日 23:59)

解答过程中请写出必要的计算和证明过程

**Q1.** (10 + 10 = 20 分)

1. 对于递归式  $T(n) = 4T(n/2) + n^2 \log^2 n$ , 给出  $T(n)$  的一个渐近紧确界;
2. 对于递归式  $T(n) = T(2n/3) + 2^{\lceil \log n \rceil}$ , 给出  $T(n)$  的一个渐近紧确界。

**Q2.** (10 + 10 + 10 + (10) = 30 + (10) 分)

问题背景: 假设我们有一个大型音视频文件, 需要对其进行编码处理。为了提高编码效率, 我们采用分治法将文件分割成较小的部分进行处理, 然后合并这些编码后的部分。具体策略如下:

将文件分为 **2** 个大小为原文件的  $\frac{1}{4}$  的子文件和 **3** 个大小为原文件的  $\frac{1}{6}$  的子文件。对这些子文件分别进行递归编码处理。合并这些处理后的子文件, 合并过程所需时间为  $n \log n$ ; 对于大小不超过 1 的文件需要花费 1 单位的时间处理。请回答以下问题:

1. 写出音视频文件的编码处理时间的递推公式  $T(n)$ ; 假设文件的初始大小为  $n = 24$ , 计算编码处理所需的总时间  $T(24)$ ;
2. 试编写程序分别计算  $T(100)$ ,  $T(1000)$ ,  $T(10000)$ , 若已知  $T(n)$  的时间复杂度为  $\theta(n^a \log^b x)$ , 尝试猜测  $T(n)$  的形式;
3. (使用替代法或递归树法) 分析并给出编码处理时间  $T(n)$  的渐近时间复杂度;
4. \* (参考教材第 4 章注记, 使用 Akra-Bazzi 定理) 分析并给出编码处理时间  $T(n)$  的渐近时间复杂度。

**Q3.** (20 分)

由于 MAX-HEAPIFY 的最后一行的递归调用可能会损失效率, 请用循环控制取代递归, 重写 MAX-HEAPIFY 代码为 NEW-MAX-HEAPIFY, 并给出 MAX-HEAPIFY 和 NEW-MAX-HEAPIFY 的时间复杂度。

```

1 MAX-HEAPIFY(A, i)
2     l = LEFT(i)

```

```

3     r = RIGHT(i)
4     if l <= A.heap-size and A[l] > A[i]
5         largest = l
6     else largest = i
7     if r <= A.heap-size and A[r] > A[largest]
8         largest = r
9     if largest != i
10         exchange A[i] with A[largest]
11     MAX-HEAPIFY(A, largest)

```

**Q4.** (15 分)

试证明：在一个随机输入数组上，对于任何常数  $0 < \alpha \leq 1/2$ ，**Partition** 产生比  $1 - \alpha : \alpha$  更平衡的划分的概率约为  $1 - 2\alpha$ 。

**Q5.** (15 分) 在  $[2, 3, 3, 5, 6, 7, 1]$  数组上对于不同基准选取策略，应用快速排序。分别写出以最右边为基准、三数取中、最均衡划分的快速排序步骤。（三数取中选取最左边、最右边、最中间三个值的中位数作为基准；最均衡划分选取使得划分最均衡的数作为基准）