

计算机学院《算法设计与分析》

(2023 年秋季学期)

第一次作业

作业提交截止时间：2023 年 10 月 05 日 23 : 55

1 请给出 $T(n)$ 尽可能紧凑的渐进上界并予以说明 (每小题 3 分, 共 21 分)

1.

$$T(1) = T(2) = 1$$

$$T(n) = T(n-2) + 1 \quad \text{if } n > 2$$

2.

$$T(1) = 1$$

$$T(n) = T(n/2) + n \quad \text{if } n > 1$$

3.

$$T(1) = 1, T(2) = 1$$

$$T(n) = T(n/3) + n^2 \quad \text{if } n > 2$$

4.

$$T(1) = 1$$

$$T(n) = T(n-1) + n^2 \quad \text{if } n > 1$$

5.

$$T(1) = 1$$

$$T(n) = T(n-1) + 2^n \quad \text{if } n > 1$$

6.

$$T(1) = 1$$

$$T(n) = T(n/2) + \log n \quad \text{if } n > 1$$

7.

$$T(1) = 1, T(2) = 1$$

$$T(n) = 4T(n/3) + n \quad \text{if } n > 2$$

2 k 路归并问题 (19 分)

现有 k 个有序数组（从小到大排序），每个数组中包含 n 个元素。你的任务是将他们合并成 1 个包含 kn 个元素的有序数组。首先来回忆一下课上讲的归并排序算法，它提供了一种合并有序数组的算法 *Merge*。如果我们有俩个有序数组的大小分别为 x 和 y ，*Merge* 算法可以用 $O(x+y)$ 的时间来合并这两个数组。

1. 如果我们应用 *Merge* 算法先合并第一个和第二个数组，然后由合并后的数组与第三个合并，再与第四个合并，直到合并完 k 个数组。请分析这种合并策略的时间复杂度（请用关于 k 和 n 的函数表示）。（9 分）
2. 针对本题的任务，请给出一个更高效的算法，并分析它的时间复杂度。（提示：此题若取得满分，所设计算法的时间复杂度应为 $O(nk \log k)$ ）。（10 分）

3 三余因子和问题 (20 分)

定义整数 i 的“3 余因子”为 i 最大的无法被 3 整除的因子，记作 $md3(i)$ ，例如 $md3(3) = 1, md3(18) = 2, md3(4) = 4$ 。请你设计一个高效算法，计算一个正整数区间 $[A, B]$, ($0 < A < B$) 内所有数的“3 余因子”之和，即 $\sum_{i=A}^B md3(i)$ ，并分析该算法的时间复杂度。例如，区间 $[3, 6]$ 的计算结果为 $1 + 4 + 5 + 2 = 12$ 。

4 填数字问题 (20 分)

给定一个长度为 n 的数组 $A[1..n]$ ，初始时数组中所有元素的值均为 0，现对其进行 n 次操作。第 i 次操作可分为两个步骤：

1. 先选出 A 数组长度最长且连续为 0 的区间，如果有多个这样的区间，则选择最左端的区间，记本次选定的闭区间为 $[l, r]$ ；
2. 对于闭区间 $[l, r]$ ，将 $A[\lfloor \frac{l+r}{2} \rfloor]$ 赋值为 i ，其中 $\lfloor x \rfloor$ 表示对数 x 做向下取整。

例如 $n = 6$ 的情形，初始时数组为 $A = [0, 0, 0, 0, 0, 0]$ 。

第一次操作为选择区间 $[1, 6]$ ，赋值后为 $A = [0, 0, 1, 0, 0, 0]$ ；

第二次操作为选择区间 $[4, 6]$ ，赋值后为 $A = [0, 0, 1, 0, 2, 0]$ ；

第三次操作为选择区间 $[1, 2]$ ，赋值后为 $A = [3, 0, 1, 0, 2, 0]$ ；

第四次操作为选择区间 $[2, 2]$ ，赋值后为 $A = [3, 4, 1, 0, 2, 0]$ ；

第五次操作为选择区间 $[4, 4]$ ，赋值后为 $A = [3, 4, 1, 5, 2, 0]$ ；

第六次操作为选择区间 $[6, 6]$ ，赋值后为 $A = [3, 4, 1, 5, 2, 6]$ ，为所求。

请设计一个高效的算法求出 n 次操作后的数组，并分析其时间复杂度。

5 数字消失问题 (20 分)

给定一长度为 n 的数组 $A[1..n]$ ，其包含 $[0, n]$ 闭区间内除某一特定数（记做消失的数）以外的所有数字（例如 $n = 3$ 时， $A = [1, 3, 0]$ ，则消失的数是 2）。这里假定 $n = 2^k - 1$ 。

1. 请设计一个尽可能高效的算法找到消失的数，并分析其时间复杂度。（8 分）
2. 若假定数组 A 用 k 位二进制方式存储（例如 $k = 2$ ， $A = [01, 11, 00]$ 则消失的数是 10），且不可以直接访存（即不可以直接通过数组的下标访问数组的内容）。目前唯一可以使用的操作是 `bit-lookup(i, j)`，其作用是用一个单位时间去查询 $A[i]$ 的第 j 个二进制位。请利用此操作设计一个尽可能高效的算法找到消失的数，并分析其时间复杂度。（12 分）