计算机学院《算法设计与分析》
(2019 年秋季学期)

第一次作业

作业提交截止时间：2019 年 10 月 10 日 $23:59$

1 请给出 $T(n)$ 尽可能紧凑的渐进上界并予以说明，可以假定 n 是 2 的整数次幂。(每小题 3 分，共 21 分)

1.
$$T(1) = T(2) = 1$$
$$T(n) = T(n-2) + n \quad if \quad n > 2$$

2.
$$T(1) = 1$$
$$T(n) = 4T(n/2) + n \quad if \quad n > 1$$

3.
$$T(1) = 1$$
$$T(n) = 2T(n/2) + n \quad if \quad n > 1$$

4.
$$T(1) = 1$$
$$T(n) = 2T(n/2) + n\log n \quad if \quad n > 1$$

5.
$$T(1) = 1$$
$$T(n) = 2T(n/2) + n^2 \quad if \quad n > 1$$

6.
$$T(1) = 1$$
$$T(n) = 3T(n/2) + n \quad if \quad n > 1$$

7.
$$T(1) = 1$$
$$T(n) = T(n/2) + n\log n \quad if \quad n > 1$$

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 $Merge$。如果我们有两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 战线补给问题 (20 分)

现有 2^l 个堡垒组成一条战线，编号为 $1,2,3,\cdots,2^l$。其中每个堡垒都可能有一个或多个士兵驻扎，也可能没有任何士兵驻扎。已知士兵共有 n 个，且第 i 个士兵驻扎在编号为 a_i 的堡垒中。现请你用最小的费用给整条战线提供补给（即是说，为区间 $[1,2^l]$ 中的所有堡垒提供补给）。为战线 $[1,2^l]$ 提供补给可以按照下述两种方式进行：

1. 若当前区间组成的战线中还剩余至少两个堡垒，可以将该区间**均分**为左右两段，并分别为其提供补给。所需的总费用为补给这两段战线的费用之和。
2. 直接为当前区间组成的战线提供补给。若这段战线中的所有堡垒均没有任何士兵驻扎，则所需的费用为 A；否则，费用为 $num\times len\times B$。其中 A,B 为给定的常数，num,len 分别为这段战线中士兵的总数以及堡垒的总数。

例如，现有四个堡垒，编号为 $1,2,3,4$。只有一个士兵，其驻扎在 2 号堡垒。可选的补给方式有多种：

一种方式为直接为整条战线 $[1,4]$ 提供补给，所需费用为 $1\times 4\times B$，其中 1 表示当前中共有 1 名士兵，4 表示当前区间中共有 4 个堡垒；

另外一种可行的方式是将该区间平均分为两段 $[1,2]$ 和 $[3,4]$。并分别为这两段提供补给，和第一种方式类似，区间 $[1,2]$ 所需的代价为 $1\times 2\times B$，其中 $1,2$ 分别表示当前区间的士兵个数和堡垒个数。而区间 $[3,4]$ 由于没有任何士兵驻扎，所需的费用为 A。因此，这种补给方式的总费用为 $A+2B$。

当然也可以将区间 $[1,2]$ 或 $[3,4]$ 继续均分并分别进行补给。

请设计一个高效的算法计算为整条战线体提供补给所需的最小费用，并尽可能准确地分析该算法的时间复杂度。

4 区间计数问题 (20 分)

给定一个包含 n 个元素的数组 $A=[a_1,a_2,\cdots,a_n]$。对数组 A 中的任意区间 $[l,r](1\le l\le r\le n)$，该区间的和可表示为 $S_{[l,r]}=\sum_{i=l}^{r}a_i$。

请设计一个高效的分治算法统计有多少个区间 $[l,r]$ 满足：$X\le S_{[l,r]}\le Y$（X,Y 为给定的常数）。并分析该算法的时间复杂度。

5 向量的最小和问题 (20 分)

给定 n 个二维向量 $\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n$。每一个向量 $\mathbf{v}_i=(x_i,y_i)$ 都可以变换为如下四种形式：

1. $\mathbf{v}_i^1=(x_i,y_i)$
2. $\mathbf{v}_i^2=(-x_i,y_i)$
3. $\mathbf{v}_i^3=(x_i,-y_i)$

4. $\mathbf{v}_i^4 = (-x_i, -y_i)$

请你设计一个高效的算法从 n 个向量中找出两个向量，使得他们以某种形式相加后的模长最小。换言之，请找出两个向量 $\mathbf{v}_i, \mathbf{v}_j (1 \le i, j \le n$ 且 $i \neq j)$，以及两个整数 $k_1, k_2 (1 \le k_1, k_2 \le 4)$，使得 $||\mathbf{v}_i^{k_1} + \mathbf{v}_j^{k_2}||_2$ 最小。此外，请分析该算法的时间复杂度。