

2021 春季学期 算法设计与分析 课后作业

任课教师: 陈宇

作业提交注意事项

- 格式要求: 电子版, 格式统一为 pdf, 推荐使用 LaTeX 排版. 不接受手写扫描, 手写文字识别不清会有误判可能.
- 道德要求: 独立完成, 不可以合作讨论, 严禁抄袭!!!
- 提交方式: 将作业发送至助教涂彬彬老师的邮箱 `tubinbin@mail.sdu.edu.cn`, 邮件的主题格式请设置为“姓名 + 学号 + 2021 算法作业”
- 最后提交时间: 待通知

1 算法分析技术

1.1 假设 f 和 g 是定义在自然数集合上的函数, 若对某个其他函数 h 有 $f = O(h)$ 和 $g = O(h)$ 成立, 那么证明 $f + g = O(h)$

1.2 设 x 为实数, n, a, b 为整数, 证明下述性质:

$$\left\lceil \frac{\left\lfloor \frac{n}{a} \right\rfloor}{b} \right\rceil = \left\lceil \frac{n}{ab} \right\rceil, \quad \left\lfloor \frac{\left\lceil \frac{n}{a} \right\rceil}{b} \right\rfloor = \left\lfloor \frac{n}{ab} \right\rfloor$$

1.3 对于下面每个函数 $f(n)$, 用 Θ 符号表示成 $f(n) = \Theta(g(n))$ 的形式, 其中 $g(n)$ 要尽可能简洁. 比如 $f(n) = n^2 + 2n + 3$ 可以写成 $f(n) = \Theta(n^2)$. 然后按照阶递增的顺序将这些函数进行排列:

$$(n-2)!, 5 \log(n+100)^{10}, 2^{2n}, 0.001n^4 + 3n^3 + 1, (\ln n)^2, n^{1/3} + \log n, 3^n, \log(n!), \log(n^{n+1}), 1 + \frac{1}{2} + \cdots + \frac{1}{n}$$

1.4 求解以下递推方程:

$$\begin{cases} T(n) = T(n-1) + n^2 \\ T(1) = 1 \end{cases}$$

1.5 求解以下递推方程:

$$\begin{cases} T(n) = 9T(n/3) + n \\ T(1) = 1 \end{cases}$$

1.6 求解以下递推方程:

$$\begin{cases} T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + cn, c \text{ 为常数} \\ T(1) = 1 \end{cases}$$

2 排序类算法

2.1 设 $A = \{a_1, a_2, \dots, a_n\}$, $B = \{b_1, b_2, \dots, b_m\}$ 是整数集合, 其中 $m = O(\log n)$. 设计一个算法找出集合 $C = A \cap B$. 要求给出算法伪码描述和复杂度分析.

2.2 设 A 是 n 个数构成的数组, 其中出现次数最多的数称为众数. 设计一个算法求 A 的众数, 给出伪码和最坏情况下的复杂度.

3 分治算法

3.1 设 A 是 n 个非 0 实数构成的数组, 设计一个算法重新排列数组中的数, 使得负数都排在正数前面. 要求算法使用 $O(n)$ 的时间和 $O(1)$ 的空间.

3.2 设 S 是 n 个不等的正整数集合, n 为偶数, 给出一个算法将 S 划分为子集 S_1 和 S_2 , 使得 $|S_1| = |S_2| = n/2$, 且 $|\sum_{x \in S_1} x - \sum_{x \in S_2} x|$ 达到最大, 即使得两个子集元素之和的差达到最大.

3.3 设 A 和 B 都是从小到大已经排好序的 n 个不等的整数构成的数组, 如果把 A 和 B 合并后的数组记作 C , 设计一个算法找出 C 的中位数并给出复杂度分析.

4 贪心算法

4.1 若在 0-1 背包问题中, 各物品依重量递增排列时, 其价值恰好依递减序排列. 对于这个特殊的 0-1 背包问题, 设计一个有效算法找出最优解, 并说明算法的正确性.

4.2 将最优装载问题的贪心算法推广到两艘船的情形, 贪心算法仍然能产生最优解么? 若能, 给出证明. 若不能, 请给出反例.

4.3 设 $\Gamma = \{1, \dots, n\}$ 是 n 个字符的集合. 证明关于 Γ 的任何最优前缀码可以表示长度为 $2n - 1 + n \lceil \log n \rceil$ 位的编码序列. (提示: 先考虑树结构的编码, 再考虑叶结点对应字符的编码)

5 动态规划

5.1 图书馆大门前有 n 级台阶, 你每次跨上 1 级或者 2 级, 请问等上 n 级台阶总共有多少种不同的方法? 设计一个算法求解上述问题, 尝试写出公式, 说明算法设计思想和时间复杂度.

5.2 n 种币值 v_1, v_2, \dots, v_n 和总钱数 M 都是正整数. 如果每种币值的钱币至多使用 1 次, 问: 对于 M 是否可以有一种找零钱的方法? 设计一个算法求解上述问题. 说明算法设计思想, 分析算法最坏情况下的时间复杂度.

5.3 设 P 是一台高性能服务器, $T = \{1, 2, \dots, n\}$ 是 n 个计算任务集合, a_i 表示任务 i 所申请的计算资源. 已知服务器的最大计算资源是正整数 K . 请确定 T 的一个子集 S , 使得 $\sum_{i \in S} a_i \leq K$, 且 $K - \sum_{i \in S} a_i$ 的值达到最小. 请设计一个算法求解 S , 并分析最坏情况下的时间复杂度.

6 编程实践

You are recommended to practice the theoretical stuffs you learned in the class on POJ (<http://poj.org>).

6.1 Divide-and-Conquer

- Median: POJ-2388
- Ultra-QuickSort: POJ-2299

6.2 Greedy Algorithm

- SSSP (Dijkstra, Bellman-Ford, Floyd): POJ-1860, POJ-3259, POJ-1062, POJ-2253, POJ-1125, POJ-2240
- MST (Prim, Kruskal): POJ-1789, POJ-2485, POJ-1258, POJ-3026

6.3 Dynamic Programming

- Largest common string: POJ-1934
- Longest increasing subsequence: POJ-3903