

1. (20分) 根据增长的阶来对下面8个函数排序, 并对排序结果给出必要的解释。如果 $f(n) = O(g(n))$, 那么 $f(n)$ 应该在 $g(n)$ 的前面。如果两个函数是渐近相等的, 即 $f(n) = \Theta(g(n))$, 那么将它们放在同一个集合中。比如给定函数 $n, \log n, n+4, n^2$, 排序结果是 $\log n, \{n, n+4\}, n^2$ 。

$$\log^3 n, n^{\log \log n}, n^{1+1/\log n}, 4^{\log n}, n!, 2^{\sqrt{2 \log n}}, n^2, (1.1)^n$$

2. (32分) 求解下列递归式。对于每一个递归式, 可以假设 $T(1) = T(0) = c$, 其中 c 是一个正常量。

a) $T(n) = 3T(n/2) + n$

b) $T(n) = 2T(n-1) + 2$

c) $T(n) = T(3n/4) + T(n/4) + n$

d) $T(n) = T(7n/8) + n$

3. (28分) 给定全序空间中的 n 个元素, 现在需要找出最大和第二大的元素, 可以使用的操作只有比较操作 $<$, 即给定两个元素 a 和 b , 你可以知道 $a < b$ 的值是真还是假。

a) 设计一个算法, 可以使用不多于 $2n - 3$ 次比较操作找到最大和第二大的元素。

b) 当 n 是2的整数次幂时, 设计一个分治算法, 使用不超过 $\frac{3n}{2} - 2$ 次比较操作找到最大和第二大的元素。分析你的算法需要的比较次数。

4. (20分) 给定一个整型数组 $A[1..n]$, 其中相邻的元素最多差1, 且 $A[1] < A[n]$ 。比如 $[10, 9, 9, 8, 9, 10, 10, 11, 12, 13, 14, 15, 14, 13, 13, 13, 12]$ 。给定满足条件 $A[1] < t < A[n]$ 的整数 t , 设计一个尽可能高效的算法, 寻找 t 出现的位置。如果 t 出现多次, 返回任意一个正确的位置即可。如果 t 不存在, 返回 -1 。分析你的算法的时间复杂度。