

算法设计与分析

第 1 章作业参考答案

作业情况

教材：算法设计技巧与分析 [沙特]M. H. Alsuwaiyel

题目编号：1.13, 1.14, 1.15, 1.38

题目范围：算法分析基本概念

邮箱：wjyyyl@126.com

习题 1.13

用 true 或 false 填空。

$f(n)$	$g(n)$	$f = O(g)$	$f = \Omega(g)$	$f = \Theta(g)$
$2n^3 + 3n$	$100n^2 + 2n + 100$			
$50n + \log n$	$10n + \log \log n$			
$50n \log n$	$10n \log \log n$			
$\log n$	$\log^2 n$			
$n!$	5^n			

答案：

$f(n)$	$g(n)$	$f = O(g)$	$f = \Omega(g)$	$f = \Theta(g)$
$2n^3 + 3n$	$100n^2 + 2n + 100$	false	true	false
$50n + \log n$	$10n + \log \log n$	true	true	true
$50n \log n$	$10n \log \log n$	false	true	false
$\log n$	$\log^2 n$	true	false	false
$n!$	5^n	false	true	false

解析：对所有函数，在描述复杂度时一般只取增长最快的，因此对每一行，我们分别有：

$f(n)$	$g(n)$	$f = O(g)$	$f = \Omega(g)$	$f = \Theta(g)$
$\Theta(n^3)$	$\Theta(n^2)$	false	true	false
$\Theta(n)$	$\Theta(n)$	true	true	true
$\Theta(n \log n)$	$\Theta(n \log \log n)$	false	true	false
$\Theta(\log n)$	$\Theta(\log^2 n)$	true	false	false
$\Theta(n!)$	$\Theta(5^n)$	false	true	false

再去比较函数之间上下界关系，填出是否满足上述三个关系。

习题 1.14

用 Θ 符号表示下列函数：

(a) $2n + 3 \log^{100} n$

(b) $7n^3 + 100n \log n + 3n$

(c) $3n^{1.5} + (\sqrt{n})^3 \log n$

(d) $2^n + 100^n + n!$

答案：

(a) $\Theta(n)$

(b) $\Theta(n^3)$

(c) $\Theta(n^{1.5} \log n)$

(d) $\Theta(n!)$

解析：在分析时间复杂性时，只考虑增长最快的（见例 1.6），根据 1.8.6 节，有

$$1 \prec \log \log n \prec \log n \prec \sqrt{n} \prec n^{3/4} \prec n \prec n \log n \prec n^2 \prec 2^n \prec n! \prec 2^{n^2}$$

根据这一规则，取增长最快的一项放入 Θ 函数中即可。

习题 1.15

用 Θ 符号表示下列函数：

(a) $18n^3 + \log n^8$

(b) $(n^3 + n)/(n + 5)$

(c) $\log n^2 + \sqrt{n} + \log \log n$

(d) $n!/2^n + n^{n/2}$

答案：

(a) $\Theta(n^3)$

(b) $\Theta(n^2)$

(c) $\Theta(\sqrt{n})$

(d) $\Theta(n!/2^n)$

解析：原理同习题 1.14。

(d) 可用比值法求出两者中增长较快的一项。

$$f(n) = n!/2^n, g(n) = n^{n/2}$$

$$\begin{aligned}\frac{f(n)}{g(n)} &= \frac{n!/2^n}{n^{n/2}} \\ &= \frac{n!}{2^n \cdot n^{n/2}} \\ &\approx \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{2^n \cdot n^{n/2}} \\ &= \frac{\sqrt{2\pi n} \cdot n^{n/2}}{(2e)^n} \\ \left(\frac{f(n)}{g(n)}\right)^2 &= \frac{2\pi n^{n+1}}{(2e)^{2n}} \\ &= \frac{2\pi n^{n+1}}{(4e^2)^n}\end{aligned}$$

由于 $n^n \succ (4e^2)^n$, 所以 $f(n) \succ g(n)$, 原函数 $= \Theta(n!/2^n)$

习题 1.38

令 S 为一个 n 个正整数的集合, n 为偶数。请设计一个有效算法将 S 分成两个子集 S_1 和 S_2 , 使每个子集中有 $n/2$ 个元素, 且 S_1 中所有元素的和与 S_2 中所有元素的和的差最大, 这个算法的时间复杂性是什么?

答案: $\Theta(n \log n)$ (或使用插入排序, $\Theta(n^2)$)

解析: 为了使 S_1 中所有元素的和与 S_2 中所有元素的和的差最大, 应让 S_1 和 S_2 分别为集合 S 中最大和最小的 $\frac{n}{2}$ 个数, 所以需要进行排序, 为使算法有效, 使用 BOTTOMUPSORT 算法, 该算法的时间复杂度为 $\Theta(n \log n)$

这里提供一种求 k -th element 的 $O(n)$ 做法, 找到中位数后即可分出 S_1, S_2 , 有兴趣的同学可以参考:

<https://www.zhihu.com/video/1490365234607456256>

教材版本不对的同学, 修订版

1.13 题参考答案: (不唯一)

(a)[1, 2, 3, 4, 5, 6, 7, 8]

(b)[1, 5, 3, 7, 2, 6, 4, 8]

1.38 题参考: $\Omega(n^2)$, 对于每项, 用 $\Omega(n)$ 时间来做乘法, 共 n 项, 因此是 $\Omega(n^2)$;

$O(n)$, 从低项到高项, 利用 $x^i = x^{i-1} \cdot x$, 每次不用重复计算, 因此是 $O(n)$ 。

总结

题目编号: 1.13, 1.14, 1.15, 1.38

日期: 2023 年 3 月 9 日

批改人: 王骏骁

邮箱: wjyyy1@126.com

各位同学写题目注意过程, 可以写得简洁, 但最好要有。

习题 1.13: 两个函数关系之间的 O, Ω, Θ , 要么满足 1 个, 要么满足 3 个, 不会有 0 个或两个。有同学认为只能满足一个关系, 课下需要再巩固相关知识。

习题 1.15: 可以用斯特林公式算一下 d 题, 确定哪个式子决定了函数的主要复杂性。c 题有部分同学未分清 $\log^2 n$ 和 $\log n^2$, $\log n^2$ 是等于 $2 \log n$ 的。而 $\log \log n$ 是对 $\log n$ 再取对数, 所以更小。

习题 1.38: 可以用 BOTTOMUPSORT, 也可以用 INSERTSORT。取决于排序时间, 复杂度为 $\Theta(n^2)$ 或者 $\Theta(n \log n)$ 。

受刘锦桦、张睿栩、陈茜等同学“中位数”启发, 解析中提供一种求 k -th element 的 $O(n)$ 做法, 有兴趣的同学可以参考, 但需要注意, 不能称之为“排序”。因此, 写了“排序”但分析复杂度为 $O(n)$ 的同学会被判错。