# 引言和渐近分析

王畅

04833050/04832580 算法设计与分析
2022 年 2 月 25 日

相较于程设、数算等课程, 更重视证明.

- Operating system: “However, you are in an operation system class, not theory or operations research; no proofs are allowed.”

- Algorithm design: “It is strictly forbidden to state something without a proof.”

| 时间 | 概观 | 课本章节 | 评论 (仅供参考) |
| --- | --- | --- | --- |
| 3.28 及之前 | 各种求解策略 | 一至五章 | 相对熟悉, 关注证明和分析方法, 不熟悉动态规划的熟悉之 |
| 3.30 至 4.25 | 线性规划和网络流 | 六、七章 | 较新的话题, 课本可能看不太懂, 掌握几何直观和基本想法 |
| 4.27 至 5.18 | 下界问题 | 八、九章 | 技巧 (尤其是归约) 比较灵活, 需要多做题目 |
| 5.23 至 5.25 | 近似算法 | 第十章 | 主要是证明近似比的技巧 (从较小的观察出发) |
| 6.1 至 6.6 | 随机算法和在线算法 | 第十一章 | 概率论的方法: 分析错误概率以及期望的线性性 |

对证明的要求总体上是逐渐提高的

**算分小班14安排**

扫一扫二维码打开或分享给好友

检查一下是否恰有 2 次回课、一组论文和一组复习.

# 小班助教是谁?

- 王畅
  - 计算机系 2018 级本科生
  - 电邮: `wchang@pku.edu.cn`
  - 微信: wangchang327
- 批改作业、讨论和求解各种题目、书中/论文证明的答疑和讨论、课程项目的技术性问题
- 无法回答和考试、评分有关的问题, 这些问题都以任课老师/主管老师为准, 助教的任何评论都是无效的 (只代表个人).

# 关于作业

- 作业评分标准是由大班助教发布的. (建议: 论述要严密.)
- 推荐使用 Markdown 或者 LaTeX; 可以拍照, 但字迹需要清晰, 上传时应旋转好.
- (建议: You must acknowledge your sources and collaborators.)

# 问题的分类和解决

- 无法用算法解决的问题 $\rightsquigarrow$ 什么是算法?
  - 解决问题的数学上良定的一系列机械的步骤
  - 例: The mortality matrix problem
- 可以用算法解决但 (很可能) 无法高效解决的问题 (本课程后半部分)
  - 例: 旅行商问题
- 可以高效解决的问题 $\rightsquigarrow$ 什么是高效? 最高效能有多高效?
  - 一般定义为多项式时间内可求解. $O(1.0001^n)$ v.s. $O(n^{10000})$?
  - 参看 `https://cstheory.stackexchange.com/questions/6660/polynomial-time-algorithms-with-huge-exponent-constant`
  - 最坏情况? 平均情况? 怎么平均? 分析方法? (本课程前半部分)

# 输入规模

- 判定 $n \in \mathbb{N}$ 是否为素数, 采用试除法, 最坏情况的时间复杂度为 $\sqrt{n}$.
- 李雷说: 因为 $\sqrt{n} = o(n)$, 因此这是多项式时间的算法, 对吗?
- 基本操作不是针对一个一个的元素. 问题的输入规模是 $\log n$.
  - 输入规模的定义应当具有某种良定性.

# 一些平凡的注解

- 课本中的例 1.7, $\left(\sum_{0\le k\le n} x^k\right)' = \left(\sum_{0\le k\le n} kx^{k-1}\right)$ 由此也可以求解.
- 另外一个技巧 (Knuth) 如下, 求 $\sum_{1\le k\le n} H_k$, 考虑 $S_n = \sum_{k=1}^{n} kH_k$,

$$
\begin{aligned}
S_n + (n+1)H_{n+1} = 1 + \sum_{2\le k\le n+1} kH_k &= 1 + \sum_{1\le k\le n} (k+1)\left(H_k + \frac{1}{k+1}\right) \\
&= n + 1 + S_n + \sum_{1\le k\le n} H_k,
\end{aligned}
$$

  推出 $\sum_{1\le k\le n} H_k = (n+1)(H_{n+1} - 1) \sim n\log n$. 此法适用于上一例.
- 其他: 裂项 $\rightsquigarrow$ 有限微积分、Abel 变换.

# BIG OMICRON AND BIG OMEGA AND BIG THETA

Donald E. Knuth
Computer Science Department
Stanford University
Stanford, California 94305

Most of us have gotten accustomed to the idea of using the notation $O(f(n))$ to stand for any function whose magnitude is upper-bounded by a constant times $f(n)$, for all large $n$. Sometimes we also need a corresponding notation for lower-bounded functions, i.e., those functions which are at least as large as a constant times $f(n)$ for all large $n$. Unfortunately, people have occasionally been using the O-notation for lower bounds, for example when they reject a particular sorting method "because its running time is $O(n^2)$." I have seen instances of this in print quite often, and finally it has prompted me to sit down and write a Letter to the Editor about the situation.

# 渐近分析

- 有关 big-$O$ 记号的一点说明: 对于多元函数讲严格的渐近式在一定意义上是不可能的. 一般都视作单变量函数, 比如指 $n \to \infty$ 或者 $n \to 0^+$ 而其他变量固定, 或者只是为了表明省略常数. 例如 $O(n^k)$ 应理解为 $k$ 为常数, $n \to \infty$.
- 参看 `https://people.cs.ksu.edu/~rhowell/asymptotic.pdf`
- 渐近分析的缺陷: 忽略常数, 有可能和实用存在差距——“具有一定的欺骗性”.

# 两个性质

**定理 1 (Stirling)**

$$n! = \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n \mathrm{e}^{\lambda_n}, \quad \frac{1}{12n+1} < \lambda_n < \frac{1}{12n}.$$

例如 $\binom{n}{k} = \dfrac{n^k \mathrm{e}^{-\frac{k^2}{2n}-\frac{k^3}{6n^2}}}{k!}(1+o(1))$.

**定理 2 (Euler)** 设 $a, b \in \mathbb{Z}$ 且 $f(x)$ 充分光滑, 则

$$\sum_{n=a+1}^{b} f(n) = \int_a^b f(t)\,\mathrm{d}t + O\left(\int_a^b |f'(t)|\,\mathrm{d}t\right)$$

$$= \int_a^b f(t)\,\mathrm{d}t + \frac{1}{2}f(b) - \frac{1}{2}f(a) + O\left(\int_a^b |f''(t)|\,\mathrm{d}t\right).$$

# log?

**问题 3** 设函数 $f(x), g(x): \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}_{\geq 0}$ 满足 $f(+\infty) = g(+\infty) = +\infty$, 那么 $f(x) = \Theta(g(x))$ 是 $\log f(x) = \Theta(\log g(x))$ 的 (　　) 条件.

(A) 充分不必要

(B) 必要不充分

(C) 充分必要

(D) 不充分不必要

**解** A. 反例如 $n, n^2$ 或者 $n!, n^n$ 等等. □

# 排序

先分类, 然后写成 poly 或者 exp 的形式, 兼用取对数的方法判断.

**问题 4** 将下列函数表示成 $\Theta(\cdot)$ 记号的形式, 并据此按阶从小到大排列.

$$(n-1)! \qquad 5\log n + 100^{10} \qquad 2^{2n} \qquad 0.001n^4 + 3n^3 + 1 \qquad n^{\log\log n}$$

$$(\log n)^2 \qquad \sqrt[3]{n} + \log n \qquad 3^n \qquad \log n! \qquad (\log n)^{\log n}$$

$$\log n^{n+1} \qquad 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \qquad n \cdot 2^n \qquad 2^{2^n} \qquad n^n$$

# 排序

**解**

$$
\begin{aligned}
&5\log n + 100^{10} \sim H_n \qquad (\log n)^2 \\
&\sqrt[3]{n} + \log n \qquad \log n^{n+1} \sim \log n! \sim n\log n \qquad 0.001n^4 + 3n^3 + 1 \\
&n^{\log\log n} \sim (\log n)^{\log n} \qquad n\cdot 2^n \qquad 3^n \qquad 2^{2n} \\
&(n-1)! \qquad n^n \qquad 2^{2^n}
\end{aligned}
$$

□

属期中考第一题题型. 可以去做一下课本习题 1.16~1.18.

# 二重和

**问题 5** 以下和式出现在随机快速排序的分析中: 给定正整数 $n \geq 2$, 计算

$$S_n = \sum_{1 \leq j < k \leq n} \frac{1}{k-j}.$$

**解** 我们有

$$\sum_{1 \leq j < k \leq n} \frac{1}{k-j} = \sum_{1 < k \leq n} \sum_{1 \leq j < k} \frac{1}{k-j} = \sum_{1 < k \leq n} H_{k-1} = nH_n - n \sim n \log n,$$

$$\text{或者} = \sum_{1 \leq j < k+j \leq n} \frac{1}{k} = \sum_{1 \leq k \leq n} \sum_{1 \leq j \leq n-k} \frac{1}{k} = \sum_{1 \leq k \leq n} \frac{n}{k} - 1 = nH_n - n. \quad \square$$

# 隐函数

**问题 6** 设 $y^2 \log y = x^3$, 将 $y$ 渐近地写成 $x$ 的函数.

**解** 取对数 $3 \log x = 2 \log y + \log \log y = \Theta(\log y)$, 于是

$$y^2 = \frac{x^3}{\log y} = \Theta\left(\frac{x^3}{\log x}\right),$$

表明 $y = \Theta\left(\frac{x^{\frac{3}{2}}}{\sqrt{\log x}}\right)$. 参看 [Spe14, §2.6 节]. $\square$

# 选择参数

**问题 7** 假设你设计了一个算法解决某问题, 输入规模是 $n$. 这个算法由两部分组成: 一部分需要时间 $O(n^3/t)$, 另一部分需要时间 $O(t\log t)$, 其中 $t$ 是一个可以调节的参数. 如何选择 $t = h(n)$ 使得算法性能较好? (来源: Ryan O'Donnell)

**解** 注意到 $f(n) + g(n) = \Theta(\max\{f(n), g(n)\})$, 因此可取 $n^3/t = t\log t$, 这前面计算过: $t = \Theta\left(\frac{n^{\frac{3}{2}}}{\sqrt{\log n}}\right)$. 故算法的优化的运行时间是 $O(n^{\frac{3}{2}}\sqrt{\log n})$. □

谢谢! 以下为参考文献.

[Spe14] J. Spencer. *Asymptopia*. Vol. 71. Student Mathematical Library. American Mathematical Society, 2014. ISBN: 978-1-47-040904-3.

封面图片来源: 肥猴输出平台