

算法专题：绪论+算法分析

原文链接：[算法专题 - Jiang Zhou's blog \(feyxiang.top\)](#)

算法专题：绪论+算法分析

一：绪论

- 1.算法和算法设计
- 2.算法与程序

3.算法设计流程

二：算法分析

- 1.算法分析
- 2.算法性能分析
 - 1) 算法复杂性
 - 2) 时间复杂度和空间复杂度
 - 时间复杂度（算法需要时间资源的量）
 - 1.计算方法：
 - 2.情况
 - 空间复杂度（算法需要空间资源的量）
 - 程序空间组成
 - 1.程序运行时所需空间：
 - 2.程序p的空间需求量
 - 3.对递归算法空间复杂度的分析
 - 4.对非递归算法空间复杂度的分析
- 3.渐进分析法
- 4.NP问题，NPhard问题，P问题等

附：

一：绪论

1.算法和算法设计

- **算法是对特定问题求解**的一种描述，是**指令**的有效序列。
- **算法设计**的主要方法和基本思想为：贪心法，回溯法，递归和分治，动态规划法，分支限界法。
- **算法分类**：精确算法，启发式算法，近似算法，随机算法。
- 算法的性质：输入（0或者多），输出（至少一个），确定性，能行性，有穷性。
- 算法表示：自然语言，编程语言，伪代码。
- 常见的算法应用：搜索问题，排序问题，图论问题，组合数学问题，几何问题，数值计算问题。

2.算法与程序

- **程序是算法用某种程序设计语言的具体实现。**
- 算法具有有穷性，程序可以不满足有穷性。

Q：操作系统是程序还是算法？

Answer：操作系统是一个在无限循环中执行的程序，并不是算法。操作系统的各种任务都是单独的问题，每个问题通过特定的子程序中算法实现，子程序得到输出结果后终止。

3.算法设计流程

1. 理解问题，预测输入
2. 选择精确解或者近似解
3. 确定数据结构，选择算法
4. 描述算法，跟踪算法
5. 分析算法效率
6. 根据算法编写代码

二： 算法分析

1.算法分析

- 算法分析是对**算法的执行时间和所需空间**的估算。当算法写成程序后，可通过对**程序的性能测量**来分析算法。
- **程序的性能指：**程序运行时所需的**内存空间量**和**计算时间**，也就是系统开销和求解问题本身的开销。
- **性能评价的方法：**解析（算法课程以解析为主，也就是对程序进行分析），测量。

2.算法性能分析

1) 算法复杂性

算法复杂性依赖于问题规模，算法输入，算法本身的函数。

2) 时间复杂度和空间复杂度

时间复杂度（算法需要时间资源的量）

- 时间复杂度指程序执行所需要的时间

1.计算方法：

操作计数（operation count）：找出关键步骤的执行时间；

注：关键步骤可以选择不止一个。赋值，比较，加和乘的次数等都可供选择

程序步计数（step count）：确定程序总的执行步数。

注：

- 程序步是指语法语义的片段，由基本指令构成，比如算术类指令，数据移动指令，控制指令等。
- 执行时间独立于所选用的实例特征。

2.情况

最好情况：不常出现，不具有普遍性。

最坏情况：确定上界，更具有一般性。

平均情况：情况复杂，分析难度大。

注：运算时注意区分元素成功查找的平均比较次数，平均比较次数（考虑失败的比较）。

空间复杂度（算法需要空间资源的量）

- **算法的空间复杂度**指算法执行时需要的存储空间。
- **程序的空间复杂度**指程序运行时所需的**内存空间**大小和**实例特征**的函数关系。

程序空间组成

1.程序运行时所需空间：

指令空间（编译程序后指令的存储空间）。

数据空间（常量和简单变量的所需空间，**复合变量**比如数组链表树图等所需空间）

环境栈空间（保存函数返回时恢复运行所需要的信息）

2.程序p的空间需求量

- 组成：常量（与实例无关）+可变部分（与实例有关）

通常忽略和 实例特征无关的空间需求量，一般复合变量是与实例特征相关性最高）

3.对递归算法空间复杂度的分析

计算递归算法的空间复杂度通常涉及到分析递归调用所使用的内存。空间复杂度表示算法在执行过程中所需的额外内存空间，除了输入数据本身。

在递归调用中，主要的内存使用包括递归调用栈和递归函数自身所使用的内存。

步骤如下：

1. **分析递归深度：** 首先，确定递归算法的递归深度，也就是递归函数 被连续调用的次数。这将直接影响到递归调用栈的深度。
2. **分析递归函数的内存消耗：** 对于每次递归调用，分析递归函数自身所使用的内存。这包括**函数局部变量、参数、返回地址**等。递归函数的内存消耗通常与函数的代码和数据结构有关。
3. **计算总体空间复杂度：** 将递归深度乘以每次递归调用的内存消耗，即可得到总体的空间复杂度。

4.对非递归算法空间复杂度的分析

分析与实例特征有关的数据结构的大小。

3.渐进分析法

忽略常数因子和低阶项，关注算法复杂度的增长量级，这种分析叫做**渐进分析**。

渐进符号分为三种：

渐进等于 Θ （渐进紧界），渐进大于 O （渐进上界），渐进小于 Ω （渐进下界）。

以渐进大于为例讲解

当 $f(n)=O(g(n))$ ，当且仅当存在正的常数 c 和 n_0 ，使得对于所有的 $n \geq n_0$ ，有 $f(n) \leq c * g(n)$ 。

也就是对于上限比较松散。

eg: $f(n)=n^2+6n+10$

那么它的渐进上界O可以是 n^2 , n^3 , n^4 ……

渐进下界 Ω 可以是 $n^2, n, 1$

渐进等于 Θ 可以是 n^2 .

4.NP问题, NPhard问题, P问题等

略。

附：

1.内容参考

佟鑫宇老师 天津大学智能与计算学部 2023秋 算法设计与分析 ppt

2.

考试考察重点：时间复杂度，渐近分析法。

3.

全文同样包含个人的主观理解，如有错误，欢迎访问[原文链接](#)指正。