S1.2 算法分析

Hahabula

2025-02-12

目录

分析算法主要度量的是计算时间，在算法分析中假定一种通用的单处理计算模型-随机访问器**(random-access machine, RAM)**来作为实现的技术，算法可以通过计算机程序来实现。在 RAM 模型中，指令一条接一条地执行，没有并发操作。在设计算法是应当采取真实计算机时如何设计的，RAM 就如何设计的准则，即 RAM 模型包括计算机中的常见指令：

- 算术指令：如加法、减法、乘法、除法、取余、向下取整、向上取整；
- 数据移动指令：装入、存储、复制；
- 控制指令：条件与无条件转移、子程序调用与返回。

1 插入排序算法分析

1. **输入规模**的最佳概念依赖于研究的问题，在插入排序算法中是指待排序数组的规模 n。
2. 一个算法在特定输入上的**运行时间**是指执行的基本操作数或步数，假定第 i 行的每次执行时间都是常量时间。

INSERTION-SORT(A)	代价	次数
1 **for** $j = 2$ **to** $A.length$	c_1	n
2 $key = A[j]$	c_2	$n-1$
3 // Insert $A[j]$ into the sorted sequence $A[1..j-1]$.	0	$n-1$
4 $i = j - 1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 $A[i+1] = A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7 $i = i - 1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8 $A[i+1] = key$	c_8	$n-1$

Figure 1: 插入排序算法的运行时间图

- 上图中的t_j表示对那个值第 5 行执行 while 循环测试的次数。

- for 循环的本质上是判断+递增/递减，因此 for 循环会执行n次，即使到了$A.length$还会再循环一次进行判断而结束。

总运行时间如下：

$$T(n) = c_1 n + (c_2 + c_4)(n-1) + c_5 \sum_{j=2}^{n} t_j + c_6 \sum_{j=2}^{n} (t_j - 1) + c_7 \sum_{j=2}^{n} (t_j - 1) + c_8 (n-1)$$

1.1 最好情况

当循环到第 i 个元素时，最好情况就是其前元素小于它，则无需进入 while 循环直接来到第 $i+1$ 个元素，也即是有如下总运行时间：

$$\begin{aligned} T(n) &= c_1 n + (c_2 + c_4)(n-1) + c_5 (n-1) + c_8 (n-1) \\ &= (c_1 + c_2 + c_4 + c_5 + c_8) n - (c_2 + c_4 + c_5 + c_8) \end{aligned}$$

也即最好情况的运行时间是 n 的线性函数。

1.2 最坏的情况

最坏的情况是值每循环到第 i 个元素时，其前子序列均大于它，它需要插入到 1 的位置，则有 $\sum_{j=2}^{n} t_j = 2 + 3 + ... + n = \frac{(n-1)(n+2)}{2}$（第 2 个元素只需要交换一次但是 t_j 包括 while 的判断），则总运行时间为：

$$\begin{aligned} T(n) = {} & c_1 n + c_2 (n-1) + c_4 (n-1) + c_5 \left(\frac{n(n+1)}{2} - 1 \right) + c_6 \left(\frac{n(n-1)}{2} \right) \\ & + c_7 \left(\frac{n(n-1)}{2} \right) + c_8 (n-1) \\ = {} & a_1 n^2 + a_2 n + a_3 \end{aligned}$$

即最坏情况的运行时间是 n 的二次函数。

2 最坏情况与平均情况分析

在算法分析中往往仅关注最坏情况运行时间，其原因如下：

- 一个算法的最坏情况运行时间给出了任何输入的运行时间的一个上界；
- 对于某些算法，最坏情况经常出现；

3 增长量级

为了对运行时间进一步抽象便提出增长率/增长量级，其只考虑公式中最重要的项，例如插入排序具有最坏情况运行时间 $\Theta(n^2)$