

1.4 算法和算法分析

1.4.1 算法的定义、基本特征、描述方法

1.4.2 类C语言描述算法

1.4.3 算法的评价

1.4.3 算法的评价

刘 芳 LiuFang

算法设计的要求

正确性

可读性

健壮性

高效率

低存储量

- **正确性**是大体可以分为三个层次：
 - 对于几组输入数据能够得到满足要求的结果。
- 算法主要是为了人的阅读和交流，其次才是机器可执行性。
可读性好的设计有助于人们对算法的理解
- **健壮性**是算法对非法输入的抵抗能力。
 - 强调即使输入非法数据，算法应能加以识别并作出处理，不是产生误动作，或陷入瘫痪。

算法设计的要求

正确性

可读性

健壮性

高效率

低存储量

算法的效率（时间）和存储量（空间）是算法评价的两个重要指标，都与问题的规模有关。

- 算法的效率通常是指算法的执行时间，对于一个具体问题的解决通常有多种算法，执行时间短的算法效率就高。
- 算法的存储量通常是指算法执行过程中需要的存储空间。好的算法，需要的存储量低。

算法效率的度量

算法的执行时间

语句频度

算法的时间复杂度

算法的执行时间是指算法中所有语句执行的时间总和。

- 算法的实际执行时间与机器的硬件和系统软件等多种因素有关。
- 算法执行时间的本质因素是算法中基本语句执行的次数。

语句频度：是指该语句在一个算法中重复执行的次数。

$f(n)$

```
++X;
```

1

```
for(i=0; i < n; i++) ++X;
```

n

```
for(i=0; i < n; i++)  
    for(j=0; j < n; j++) ++X;
```

n^2

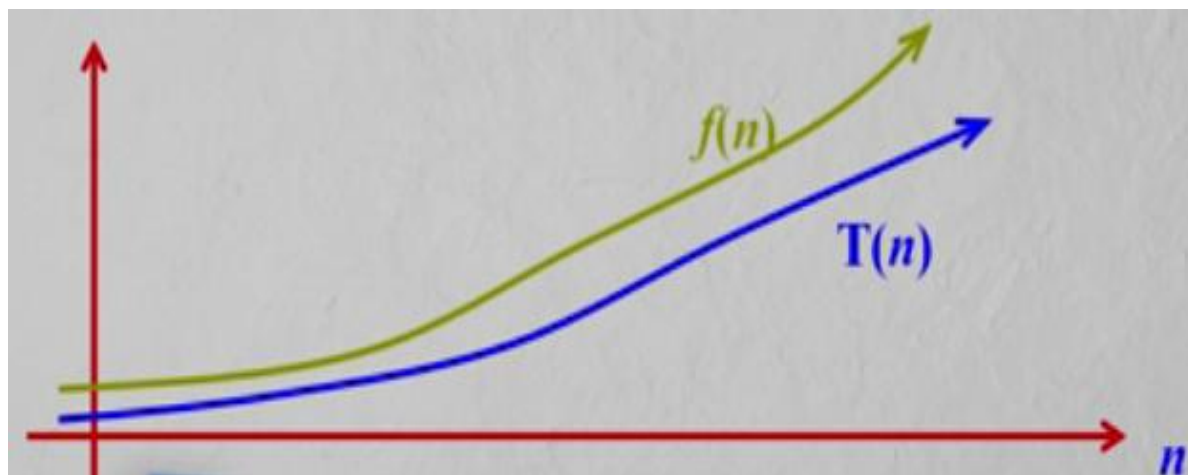
算法效率的度量

算法的执行时间

语句频度

算法的时间复杂度

- 算法的时间复杂度是指以语句频度刻画随问题规模 n 增长的函数 $f(n)$ 执行的数量级Order。
 - 记做： $T(n)=O(f(n))$
 - 表示随问题规模 n 的增长，算法的执行时间的增长率和 $f(n)$ 的增长率相同。（趋势分析）



算法效率的度量

算法的执行时间

语句频度

算法的时间复杂度

常用复杂度的阶	名称
$O(1)$	常量阶
$O(\log n)$	对数阶
$O(n)$	线性阶
$O(n \log n)$	线性对数阶
$O(n^2)$	平方阶
$O(n^3)$	立方阶
$O(n^k)$	多项式阶
$O(2^n)$	指数阶
$O(n!)$	阶乘阶

有效算法

无效算法

算法效率的度量

算法的执行时间

语句频度

算法的时间复杂度

■ 算法的时间复杂度的三种情况

- 最好情况
- 最坏情况
- 平均情况

■ 例如：冒泡排序

- 最好情况 $O(n)$
- 最坏情况 $O(n^2)$
- 平均情况 $O(n^2)$

算法存储空间的度量

算法的存储量

算法的空间复杂度

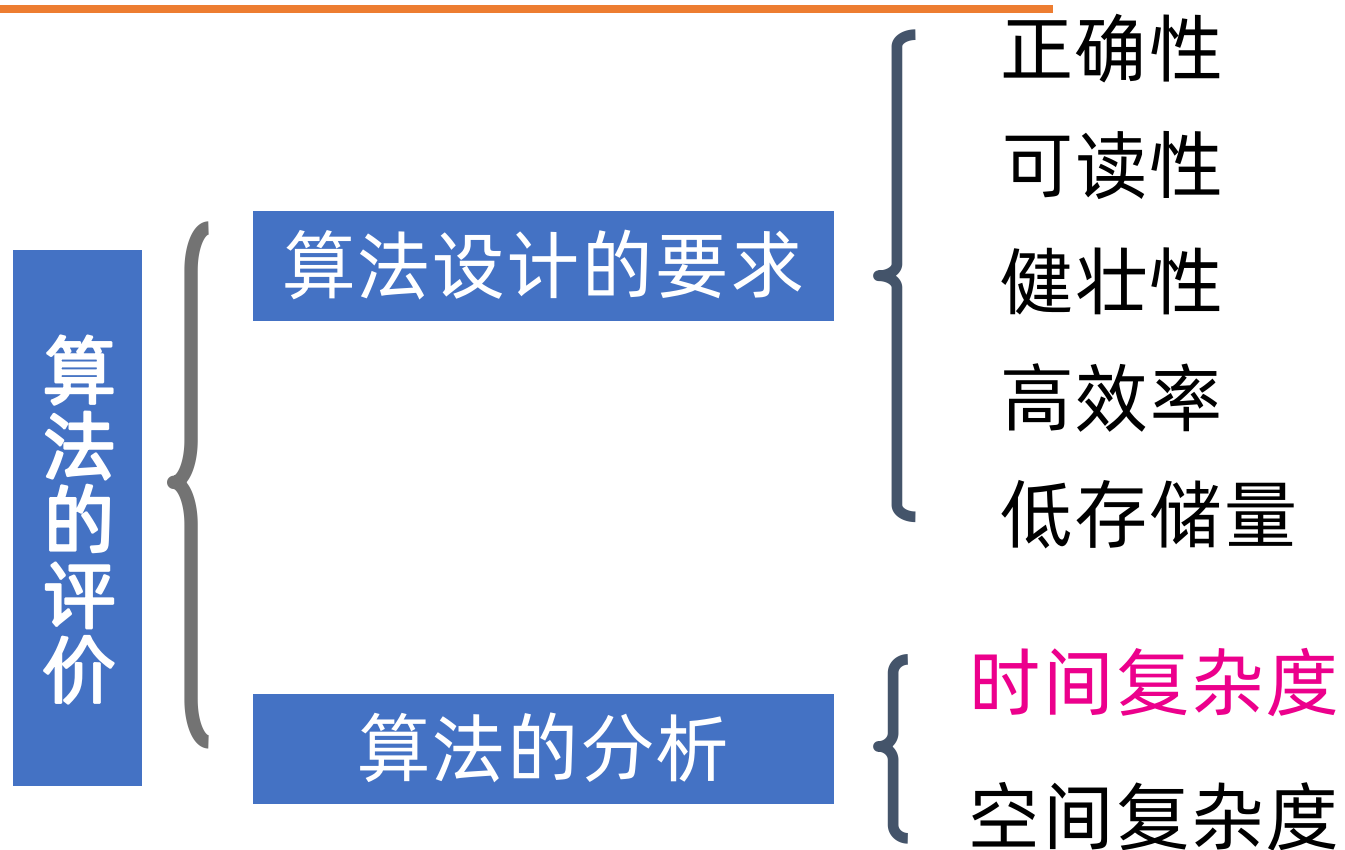
■ 算法的存储空间包括：

- 输入数据所占空间
- 程序本身所占空间
- 辅助变量所占空间

■ $S(n)=O(f(n))$

- 最好情况
- 最坏情况
- 平均情况

本节要点



感谢聆听

黑发不知勤学早,白发方悔读书迟。