

绪论

基本概念和术语

数据：
数据元素：是数据（集合）中的一个“个体”，是数据结构讨论的基本单位
关键字：能识别一个或者几个数据元素的数据项。若能起唯一识别作用，则称为“主”关键字，否则为“次”
数据对象：具有相同特性的数据元素的集合 $D = \{ \dots \}$

数据结构：是相互之间存在某种逻辑关系的数据元素的集合

数据项：其中每一个数据，是数据结构中讨论的最小单位。
数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

数据项：
数据项：其中每一个数据，是数据结构中讨论的最小单位。

算法

是为了解决某类问题而规定的一个有限长的操作序列
五个重要特性：
有效性：对于任意一组合法输入，在执行有穷的步骤之后一定能够结束。算法中的每个步骤都能在有限时间内完成
确定性：对于每种情况下所应执行的操作，在算法中都有确切的规定。使算法的每一步骤或操作都能明确地定义及如何执行。并在任何条件下，算法都只有一条执行路径。
可行性：算法中的所有操作都必须足够基本，都可以通过已经实现的原始操作运算而实现之。
有输入：作为算法加工对象的量值。通常体现为算法中的一组变量。有些输入量是算法运行过程所需输入，而有的算法原则上可以没有输入，实际上已输入算法之中。
有输出

设计算法时应考虑达到以下目标

正确性

a. 不含语法错误
b. 对于几组输入数据能够得出满足要求的结果
c. 程序对于精心选择的、典型、糟糕且带有刁钻性的几组输入数据能够得出满足要求的结果
d. 程序对于一切合法的输入数据都能得出满足要求的结果

可读性

算法主要是为了人的阅读与交流，其次才是为计算机执行
设计算法时应考虑达到的目标

健壮性

健壮性：当输入的数据非法时，算法应当能作出反映或进行相应处理，而不是产生莫名其妙的输出结果
健壮性：当输入的数据非法时，算法应当能作出反映或进行相应处理，而不是产生莫名其妙的输出结果

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

效率与程序质量需求

如何估计时间和空间复杂度

问题的规模

初始复杂度

空初始复杂度

编写程序的语言

编译程序产生的目标代码的质量

计算机执行指令的速度

输入数据所占空间

程序本身所占空间

辅助变量所占空间

两参数与问题的规模有关

事后统计法

事前分析法

缺点：1. 必须执行程序
2. 其它因素掩盖算法本质

优点：可以预先比较各种算法，以便选择最优算法

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度

如何估计时间和空间复杂度