

## 第 1 章 绪论

数据、数据元素（元素、结点、顶点、记录）、数据项

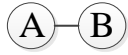
结构（关系）：逻辑结构、存储结构

用圆圈表示元素，线段表示关系。

逻辑结构：集合（元素间没有关系）、线性（元素间存在一对一的关系）、树形（元素间存在一对多的关系）、图状（元素间存在多对多的关系）

p. 7 图 1-2

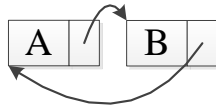
存储结构：顺序、链式  
两个有关系的元素 A 和 B



顺序存储结构示意图



链式存储结构示意图



元素+关系=数据结构

数据结构+操作=抽象数据类型

数据域+指针域（顺序实现时无）=结点

算法特性：有穷性、确定性、可行性、输入（可以没有）、输出（至少一个）

算法描述语言：程序设计语言、类程序设计语言、自然语言、图表等

好算法的目标：正确性、可读性、健壮性（稳健性、鲁棒性）、高效性

算法的效率：时间、空间

时间和空间往往是一对矛盾。

交换两个整型变量 a 和 b 的值：

(1)

c = a;

a = b;

```

b = c;
(2)
a = a + b;
b = a - b;
a = a - b;

```

与程序执行时间相关的因素：算法思想、问题规模、程序设计语言、编译程序、CPU 等

时间复杂度、空间复杂度

两个  $n \times n$  方阵相乘算法：

```

for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
        c[i][j] = 0;
        for (k = 1; k <= n; k++) {
            c[i][j] += a[i][k] * b[k][j];
        }
    }
}

```

$T(n) = t_0 + t_1n + t_2n^2 + t_3n^3 = O(n^3)$

观察当  $n$  趋向于正无穷大时，哪一项占的比例越来越大。

$O(n^3)$  读作“大  $O$   $n$  立方”、“ $n$  立方的大  $O$ ”或“ $n$  立方阶”。



算法的常见时间复杂度： $O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n)$   
 常量阶    对数阶    线性阶                     平方阶    立方阶    指数阶

最好时间复杂度、最坏时间复杂度、平均时间复杂度

冒泡排序实例：

(1)

```

60 70 50 30 80 20 10 40
60 50 30 70 20 10 40 80
50 30 60 20 10 40 70 80
30 50 20 10 40 60 70 80
30 20 10 40 50 60 70 80
20 10 30 40 50 60 70 80
10 20 30 40 50 60 70 80
10 20 30 40 50 60 70 80

```

(2)

```

10 20 30 40 50 60 70 80

```

(3)

80 70 60 50 40 30 20 10