
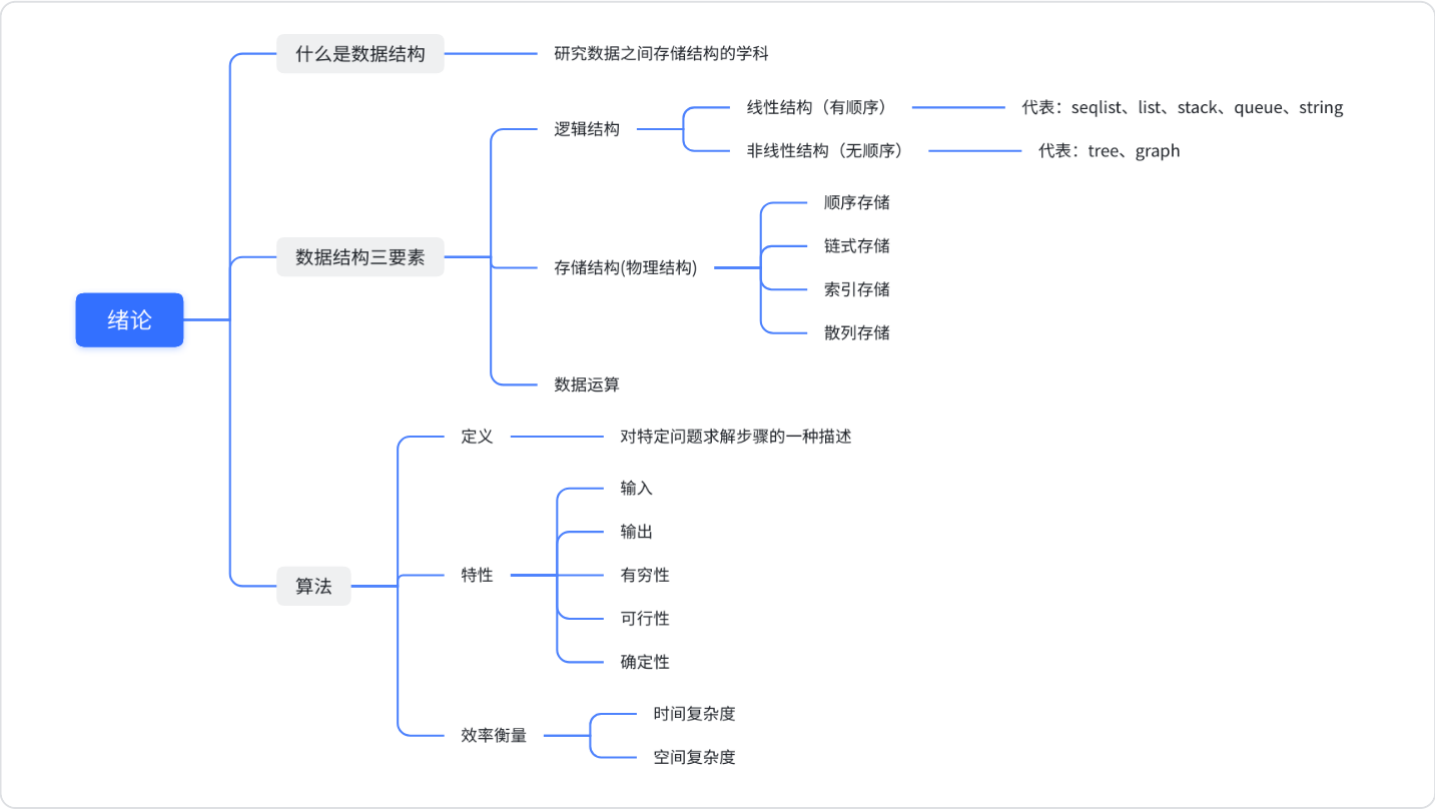


1-绪论


C生万物 ● 大道至简 ● 鲍鱼科技+v(15339278619)

1、目标

-  1、什么是数据结构
- 2、数据结构三要素
- 3、算法的定义、特性、效率衡量



2、快速学习说明:

-  1、数据结构做什么：使用不同的物理存储结构实现逻辑结构+排序，最终为搜索数据进行服务
- 2、数据结构学什么：数组+链表，所有的数据结构都是由这两种结构组合而成的

3、时间复杂度和空间复杂度后期讲解，前期可以不需要太关心，只需要有概念基础即可，没有深厚的数学功底和编程功底，考到了大概率就是靠猜了

3、需要学习的结构



- 线性结构 - SeqList、List、Stack、Queue、String
- 非线性结构 - Tree、BinTree、Graph
- 排序 - 插入排序、选择排序、交换排序、归并排序、基数排序
- 搜索 - hashTable、BST、AVL、RBTree、B、B+、T树、倒排表

4、有一个需要知道的类型

```

ADT List {
    数据对象:  $D = \{ a_i \mid a_i \in \text{ElemSet}, i = 1, 2, \dots, n, n \geq 0 \}$ 
    数据关系:  $R1 = \{ \langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D, i = 2, \dots, n \}$ 
    基本操作:
        InitList( &L )
            操作结果: 构造一个空的线性表 L。
        DestroyList( &L )
            初始条件: 线性表 L 已存在。
            操作结果: 销毁线性表 L。
        ClearList( &L )
            初始条件: 线性表 L 已存在。
            操作结果: 将 L 重置为空表。
        ListEmpty( L )
            初始条件: 线性表 L 已存在。
            操作结果: 若 L 为空表, 则返回 TRUE, 否则返回 FALSE。
        ListLength( L )
            初始条件: 线性表 L 已存在。
            操作结果: 返回 L 中数据元素个数。
        GetElem( L, i, &e )
            初始条件: 线性表 L 已存在,  $1 \leq i \leq \text{ListLength}(L)$ 。
            操作结果: 用 e 返回 L 中第 i 个数据元素的值。
        LocateElem( L, e, compare() )
            初始条件: 线性表 L 已存在, compare() 是数据元素判定函数。
            操作结果: 返回 L 中第 1 个与 e 满足关系 compare() 的数据元素的位序。若这样的数据元素不存在, 则返回值为 0。
        PriorElem( L, cur_e, &pre_e )
            初始条件: 线性表 L 已存在。
            操作结果: 若 cur_e 是 L 的数据元素, 且不是第一个, 则用 pre_e 返回它的前驱, 否则操作失败, pre_e 无定义。
        NextElem( L, cur_e, &next_e )
            初始条件: 线性表 L 已存在。
            操作结果: 若 cur_e 是 L 的数据元素, 且不是最后一个, 则用 next_e 返回它的后继, 否则操作失败, next_e 无定义。
        ListInsert( &L, i, e )
            初始条件: 线性表 L 已存在,  $1 \leq i \leq \text{ListLength}(L) + 1$ 。
            操作结果: 在 L 中第 i 个位置之前插入新的数据元素 e, L 的长度加 1。
        ListDelete( &L, i, &e )
            初始条件: 线性表 L 已存在且非空,  $1 \leq i \leq \text{ListLength}(L)$ 。
            操作结果: 删除 L 的第 i 个数据元素, 并用 e 返回其值, L 的长度减 1。
        ListTraverse( L, visit() )
            初始条件: 线性表 L 已存在。
            操作结果: 依次对 L 的每个数据元素调用函数 visit()。一旦 visit() 失败, 则操作失败。
} ADT List

```

5、数据结构的性能指标



- 时间复杂度
- 空间复杂度



这个知识点，如果在考试当中单纯考到，绝大多数就是要靠“猜”，因为难点不在于数据结构本身，而在于数学函数问题，是否能够归纳出规模函数，就是取胜的关键，这个内容将会在栈和队列知识之后讲解。

同时，这个知识点也是我们需要攻克的一个难点和关键点，因为在每年的考研综合应用题中，必然会让写出算法的时间复杂度，我们至少需要达到常见算法的复杂度计算

41. (13 分)给定一个含 $n(n \geq 1)$ 个整数的数组，请设计一个在时间上尽可能高效的算法，找出数组中未出现的最小正整数。例如，数组 $\{-5, 3, 2, 3\}$ 中未出现的最小正整数是 1；数组 $\{1, 2, 3\}$ 中未出现的最小正整数是 4。要求：

2018年

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

41. (13 分) 定义三元组 (a, b, c) (a, b, c 均为正数) 的距离 $D = |a - b| + |b - c| + |c - a|$ 。给定 3 个非空整数集合 S_1 、 S_2 和 S_3 ，按升序分别存储在 3 个数组中。请设计一个尽可能高效的算法，计算并输出所有可能的三元组 (a, b, c) ($a \in S_1, b \in S_2, c \in S_3$) 中的最小距离。例如 $S_1 = \{-1, 0, 9\}$ ， $S_2 = \{-25, -10, 10, 11\}$ ， $S_3 = \{2, 9, 17, 30, 41\}$ ，则最小距离为 2，相应的三元组为 $(9, 10, 9)$ 。要求：

2020年

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

42. (10 分) 现有 $n(n > 100000)$ 个数保存在一维数组 M 中，需要在找 M 中最小的 10 个数。请回答下列问题。

2022年

- (1) 设计一个完成上述查找任务的算法，要求平均情况下的比较次数尽可能少，简述其算法思想（不要程序实现）。
- (2) 说明你所设计的算法平均情况下的时间复杂度和空间复杂度。