

1 第一章欢迎学习玩转数据结构

1.1 1-1 欢迎学习《玩转数据结构》

数据结构研究的是数据如何在计算机中进行组织和存储，使得我们可以**高效**的获取数据或者修改数据

分类：1. 线性结构：* 数组 * 栈 * 队列 * 链表

1. 树结构：

- 二叉树
- 二分搜索树
- AVL
- 红黑树
- Treap
- Splay
- 堆
- Trie
- 线段树
- K-D 树
- 并查集
- 哈弗曼树
-

2. 图结构：

- 邻接矩阵
- 邻接表

数据结构例子：1. 数据库：—> 一个软件 —> 需要底层很多数据结构：树，哈希表等

2. 操作系统：多任务切换

1. 系统栈：递归调用
2. 优先队列：堆
 - 在多任务间比较优先级，以便进行任务切换

3. 文件压缩：哈夫曼树

4. 通讯录：Trie - 前缀树（替换链表结构-查找速度慢）

5. 游戏：寻路算法

- 图论算法 DFS：使用栈 BFS：使用队列

课程设置：Based on Java

|| 课程设置 || |—|—|—| | 数组（基础）| 二分搜索树（基础）| 并查集（竞赛）|| 栈（基础）| 堆（基础）| AVL（平衡二叉树、复杂，代码量稍大）|| 队列（基础）| 线段树（竞赛）| 红黑树（平衡二叉树、复杂，代码量稍大）|| 链表（基础）| Trie（竞赛）| 哈希表 |

不包含图：图结构使用简单的线性表就可以存储，但是图论领域以算法为主。

课程使用：LeetCode 题库

不仅单单实现，还会进行优化，揭示数据结构背后的思考。

1.2 1-2 学习数据结构（和算法）到底有没有用

现状：门槛越来越低，开发工具通过接口提供了数据结构和算法，开发者使用即可搭建上述开发工具和开发框架涉及大量数据结构

学好数据结构可以提升技术上限，在计算机科学（计算机技术）道路上走的更远

2 第二章不要小瞧数组

2.1 2-1 使用 Java 中的数组

数组三种定义方式：1. `int[] arr = new int[10];` 之后使用 for 循环为每个元素赋初值 2. `int[] arr = new int[]{100, 99, 66};` 3. `int[] arr = {100, 99, 66}` 注：1、2 中分配空间在堆中，3 分配空间在栈中

2.2 2-2 二次封装属于我们自己的数组

数组优点：随机读取，快速查询，所以数组最好应用于“索引有语意”的情况，比如索引表示学号，那么 `scores[2]` 就是获取学号为 2 的同学的分数。

但并非都是“索引有语意”最好，比如使用身份证号查询某人的工资，那么如果以身份证作索引，就要开辟很大的数组空间，其中很大一部分是浪费



的。此时可以通过一个函数对这个初始索引做进一步处理，三列在一个范围内，并且尽量避免重复，然后把这个函数的结果作为索引—类似哈希表（散列表）

数组没有语意的情况下，会有以下一些问题：1. 如何表示没有元素？2. 如何添加和删除元素？3.

```
“Java public class Array{

private int[] data;
private int size;

//构造函数，传入数组的容量 capacity 构造 Array
public Array(int capacity){
    data = new int[capacity];
    size = 0;
}
//无参构造函数，有参构造函数定以后，无参构造函数不会自动生成；默认数组容量为 capacity=10
public Array(){
    this(10);
}

//获取数组中的元素个数
public int getSize(){
    return size;
}

// 获取数组的容量
public int getCapacity(){
    return data.length;
}

// 返回数组是否为空
public boolean isEmpty(){
    return size == 0;
}
```

} “