



Фундаментальная  
библиотека



## 《数据结构》 讲义

题 目：	数据结构讲义
姓 名：	Elon Li
日 期：	完成日期

# 目录:

- 1 时间复杂度
  - 1.1 如何计算时间复杂度
- 2 线性表
  - 2.1 线性表的基本概念与实现
    - 2.1.1 线性表的定义
  - 2.2 线性表的结构体定义和基本操作
    - 2.2.1 顺序表的操作
    - 2.2.2 单链表的操作
    - 2.2.3 双链表的操作
    - 2.2.4 循环链表的操作
    - 2.2.5 逆置问题
- 3 栈和队列
  - 3.1 栈和队列的基本概念
    - 3.1.1 栈的基本概念
    - 3.1.2 队列的基本概念
  - 3.2 栈和队列的存储结构、算法与应用
    - 3.2.1 栈和队列的结构体定义
    - 3.2.2 顺序栈
    - 3.2.3 链栈
    - 3.2.4 栈的应用
    - 3.2.5 顺序队
    - 3.2.6 链队
    - 3.2.7 共享栈和双端队列
    - 3.2.8 队列配置问题
- 4 串
  - 4.1 串数据类型的定义
    - 4.1.1 串的定义
    - 4.1.2 串的存储结构
    - 4.1.3 串的基本操作
  - 4.2 模式匹配
    - 4.2.1 简单模式匹配算法
    - 4.2.2 KMP算法
    - 4.2.3 KMP算法的改进
- 5 矩阵、数组与广义表
  - 5.1 数组的定义
    - 5.1.1 矩阵的压缩存储
    - 5.1.2 特殊矩阵和稀疏矩阵
  - 5.2 广义表
- 6 树和二叉树
  - 6.1 树的基本概念
    - 6.1.1 树的定义
    - 6.1.2 树的基本术语
    - 6.1.3 树的存储结构
  - 6.2 二叉树
    - 6.2.1 二叉树的定义

- 6.2.2 二叉树的主要性质
  - 6.2.3 二叉树的遍历算法
  - 6.2.4 二叉树的遍历算法的改进
- 6.3 树与二叉树的互相转换
  - 6.3.1 树转为二叉树
  - 6.3.2 二叉树转为树
  - 6.3.3 森林转换为二叉树
  - 6.3.4 二叉树转换为森林
  - 6.3.5 树与森林的遍历
- 7 图**
  - 7.1 图的基本概念
  - 7.2 图的存储结构
    - 7.2.1 邻接矩阵
    - 7.2.2 邻接表
    - 7.2.3 邻接多重表
  - 7.3 图的遍历算法操作
    - 7.3.1 深度优先搜索遍历
    - 7.3.2 广度优先搜索遍历
  - 7.4 最小（代价）生成树
    - 7.4.1 普里姆算法和克鲁斯卡算法
  - 7.5 最短路径
    - 7.5.1 迪杰斯特拉算法
    - 7.5.2 弗洛伊德算法
  - 7.6 拓扑排序
    - 7.6.1 AOV网
    - 7.6.2 拓扑排序核心算法
  - 7.7 关键路径
    - 7.7.1 AOV网
    - 7.7.2 关键路径核心算法
- 8 排序**
  - 8.1 排序的基本概念
    - 8.1.1 排序
    - 8.1.2 稳定性
    - 8.1.3 排序算法的分类
  - 8.2 插入类排序
    - 8.2.1 直接插入排序
    - 8.2.2 折半插入排序
    - 8.2.3 希尔排序
  - 8.3 交换类排序
    - 8.3.1 起泡排序
    - 8.3.2 堆排序
  - 8.4 选择类排序
    - 8.4.1 简单选择排序
    - 8.4.2 堆排序
  - 8.5 二路归并排序
  - 8.6 基数排序
  - 8.7 外部排序

- 8.7.1 概念与流程
- 8.7.2 置换-选择排序
- 8.7.3 最佳归并排序
- 8.7.4 败者树
- 8.7.5 时间与空间复杂度相关排序

## 9 查找

- 9.1 查找的基本概念、顺序查找法、折半查找法
  - 9.1.1 查找的基本概念
  - 9.1.2 顺序查找法
  - 9.1.3 折半查找法
  - 9.1.4 分块查找法
- 9.2 树型查找法
  - 9.2.1 二叉排序树
  - 9.2.2 平衡二叉树
  - 9.2.3 红黑树
- 9.3 B-树的基本概念及其基本操作、B+树的基本概念
  - 9.3.1 B-树（B树）的基本概念
  - 9.3.2 B-树的基本操作
  - 9.3.3 B+树的基本操作
- 9.4 散列表
  - 9.4.1 散列表的概念
  - 9.4.2 散列表的建立方法以及冲突解决方法
  - 9.4.3 散列表的性能分析

## 1 时间复杂度

### 1.1 如何计算时间复杂度

## 2 线性表

### 2.1 线性表的基本概念与实现

#### 2.1.1 线性表的定义

### 2.2 线性表的结构体定义和基本操作

#### 2.2.1 顺序表的操作

#### 2.2.2 单链表的操作

#### 2.2.3 双链表的操作

#### 2.2.4 循环链表的操作

#### 2.2.5 逆置问题

## 3 栈和队列

### 3.1 栈和队列的基本概念

#### 3.1.1 栈的基本概念

#### 3.1.2 队列的基本概念

### 3.2 栈和队列的存储结构、算法与应用

#### 3.2.1 栈和队列的结构体定义

#### 3.2.2 顺序栈

#### 3.2.3 链栈

#### 3.2.4 栈的应用

#### 3.2.5 顺序队

#### 3.2.6 链队

#### 3.2.7 共享栈和双端队列

#### 3.2.8 队列配置问题

## 4 串

### 4.1 串数据类型的定义

- 4.1.1 串的定义
- 4.1.2 串的存储结构
- 4.1.3 串的基本操作
- 4.2 模式匹配**
- 4.2.1 简单模式匹配算法
- 4.2.2 KMP 算法
- 4.2.3 KMP 算法的改进

## **5 矩阵、数组与广义表**

- 5.1 数组的定义**
- 5.1.1 矩阵的压缩存储
- 5.1.2 特殊矩阵和稀疏矩阵
- 5.2 广义表**

## **6 树和二叉树**

- 6.1 树的基本概念**
- 6.1.1 树的定义
- 6.1.2 树的基本术语
- 6.1.3 树的存储结构
- 6.2 二叉树**
- 6.2.1 二叉树的定义
- 6.2.2 二叉树的主要性质
- 6.2.3 二叉树的遍历算法
- 6.2.4 二叉树的遍历算法的改进
- 6.3 树与二叉树的互相转换**
- 6.3.1 树转为二叉树
- 6.3.2 二叉树转为树
- 6.3.3 森林转换为二叉树
- 6.3.4 二叉树转换为森林
- 6.3.5 树与森林的遍历

## 7 图

### 7.1 图的基本概念

### 7.2 图的存储结构

#### 7.2.1 邻接矩阵

#### 7.2.2 邻接表

#### 7.2.3 邻接多重表

### 7.3 图的遍历算法操作

#### 7.3.1 深度优先搜索遍历

#### 7.3.2 广度优先搜索遍历

### 7.4 最小（代价）生成树

#### 7.4.1 普里姆算法和克鲁斯卡算法

### 7.5 最短路径

#### 7.5.1 迪杰斯特拉算法

#### 7.5.2 弗洛伊德算法

### 7.6 拓扑排序

#### 7.6.1 AOV网

#### 7.6.2 拓扑排序核心算法

### 7.7 关键路径

#### 7.7.1 AOV网

#### 7.7.2 关键路径核心算法

## 8 排序

### 8.1 排序的基本概念

#### 8.1.1 排序

#### 8.1.2 稳定性

#### 8.1.3 排序算法的分类

### 8.2 插入类排序

#### 8.2.1 直接插入排序

#### 8.2.2 折半插入排序

#### 8.2.3 希尔排序

### 8.3 交换类排序

#### 8.3.1 起泡排序

8.3.2 堆排序

## 8.4 选择类排序

8.4.1 简单选择排序

8.4.2 堆排序

## 8.5 二路归并排序

## 8.6 基数排序

## 8.7 外部排序

8.7.1 概念与流程

8.7.2 置换-选择排序

8.7.3 最佳归并排序

8.7.4 败者树

8.7.5 时间与空间复杂度相关排序

# 9 查找

## 9.1 查找的基本概念、顺序查找法、折半查找法

9.1.1 查找的基本概念

9.1.2 顺序查找法

9.1.3 折半查找法

9.1.4 分块查找法

## 9.2 树型查找法

9.2.1 二叉排序树

9.2.2 平衡二叉树

9.2.3 红黑树

## 9.3 B-树的基本概念及其基本操作、B+树的基本概念

9.3.1 B-树（B树）的基本概念

9.3.2 B-树的基本操作

9.3.3 B+树的基本操作

## 9.4 散列表

9.4.1 散列表的概念

9.4.2 散列表的建立方法以及冲突解决方法

9.4.3 散列表的性能分析