

07-03-04-DataStructure

Created on 20250126.

Last modified on 2025 年 2 月 22 日.

目录

Chapter 1 Basic

数据结构

线性表是一种逻辑结构。其存储结构分为顺序存储（如数组），链式存储（如链表）。

Chapter 2 数组

2.1 基础数组

$a[i] = a + i * \text{len}$; // i from 0; $a[i][j]$ 按行存: $a + (i * \text{len} + j) * \text{len}$; $a[i][j]$ 按列存: $a + (j * \text{len} + i) * \text{len}$;

5*5 的二维数组 a , 各元素 2 字节, $a[2][3]$ 行有限存, 地址? $2*5+3 = 13$, $13*2 = 26$, $a+26$;

2.2 后缀数组

Chapter 3 链表

Chapter 4 栈

Chapter 5 队列

5.1 基础队列

5.2 优先队列

Chapter 6 哈希表

6.1 哈希函数

哈希函数: $y=H(x)$, 输出长度不变; 相同输入每次得到相同输出; 输入差距小也会导致输出差距大, 输入差距大也可能导致输出相同。x 求 y 容易, y 求 x 困难。

MD5, message digest algorithm 5 SHA-1, SHA-2, source hash algorithm. MD5, SHA-1 存在安全隐患。

Chapter 7 堆

上浮和下沉，用于实现 priority queues

Chapter 8 树

8.1 二叉树

二叉树遍历: 前序、后续、中序。反向构造二叉树: 利用前 + 中, 或后 + 中遍历结果, 推出树的结构。只利用前 + 后不行。

树转二叉树: 第一个孩子在左, 兄弟都是右。

查找二叉树: 左 < 根 < 右 1) 左子树的值 < 根的值 < 右子树的值; 2) 一直向左达到最小值, 一直向右达到最大值; 3) 增加节点: 从根开始, 向末端方向, 插入值更小就左转, 否则右转, 到达末端增加一个叶子节点; 4) 删除节点 A: A 的左子树的最大节点替代删除的 A 的位置; 5) 扩展: 平衡二叉查找树; B 树 (m 个节点的形状平衡的)。

最优二叉树、哈夫曼树: 带权路径长度最小。1,2,8,4 构造哈夫曼树: step1: 1,2->3; 3,8,4; step2: 3,4->7; 7,8 so: 15 7 8 3 4 1 2 权值: $1*3+2*3+4*2+8*1=25$

线索二叉树: 前序、后续、中序, 列举各元素后, 叶子 LR 指针指向前后元素。

平衡二叉树: 任意结点左右子树深度差不大于 1。平衡度 = 左子树深度 - 右子树深度。

8.2 可持久化线段树

8.3 树链剖分

Chapter 9 图

有向图，无向图。完全图。

存储：邻接矩阵。 n 个点， $n \times n$ 。 i 到 j 有邻接边， $R_{ij}=1$ ，否则为 0。邻接表， $V1 \rightarrow [2, 6, \dots] \rightarrow [4, 1, \dots] \rightarrow [6, 50, \dots]$ // $V1$ 到 2 号结点距离 6，到 4 号结点距离 1，到 6 号结点距离 50

【遍历】深度优先，广度优先。

【拓扑排序，AOV 网络】有向边表示活动之间开始的先后关系。

【图的最小生成树，普里姆算法】留下的权值最小。树没有环路， n 个节点的树边最多 $n-1$ 个。染色红，逐个收集最短的一个元素进来。注意过程中不能形成环。

【图的最小生成树，克鲁斯卡尔算法】从最短的边开始选边。

Chapter 10 **END**