

## 07-03-Algorithm

Created on 20220605.

Last modified on 2023 年 2 月 15 日.



# 目录



# Chapter 1 计算模型合集

各种高效实用的计算模型



# Chapter 2 数据结构

## 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?  $2*5+3 = 13$ ,  $13*2 = 26$ ,  $a+26$ ;

## 线性表

## 树与二叉树

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

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。

## 图

有向图, 无向图。完全图。

存储: 邻接矩阵。n 个点,  $n*n$ 。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 3 随机算法

## 3.1 随机数生成

### 3.1.1 Mersenne Twister

梅森旋转 (Mersenne Twister, MT) 算法, 常用是伪随机数生成算法。算法描述如算法 1 所示。

---

**Algorithm 1:** Mersenne Twister

---

**Input:** the index is noted as  $x_{in}$ , the seed number is noted as seed  
**Output:** random number  $x_{out}$

```
1 Initialization:  $[w, n, m, r], a, f, (u, d), (s, b), (t, c), l, MT_0 \leftarrow \text{seed};$   
2 for  $i \leftarrow 1$  to  $n - 1$  do  
3    $MT_i \leftarrow f \cdot \{MT_{i-1} \oplus [MT_{i-1} \gg (w - 2)] + i\}$   
4 end  
5 for  $i \leftarrow 0$  to  $n - 2$  do  
6    $M_c \leftarrow$  the composition of the highest  $w - r$  bits of  $MT_i$  and the lowest  $r$  bits of  
    $MT_{i+1};$   
7    $M_c \leftarrow M_c \gg 1;$   
8   if the lowest bit of  $M_c$  is 1 then  
9      $M_c \leftarrow M_c \oplus a$   
10  end  
11   $MT_i \leftarrow MT_{i+m} \oplus M_c$   
12 end  
13  $x \leftarrow MT_{x_{in}};$   
14  $x_{out} \leftarrow x \oplus [(x \gg u) \& d];$   
15  $x_{out} \leftarrow x \oplus [(x \ll s) \& b];$   
16  $x_{out} \leftarrow x \oplus [(x \ll t) \& c];$   
17  $x_{out} \leftarrow x \oplus (x \gg l);$   
18 return  $x_{out};$ 
```

---



# Chapter 4 算法合集

一般难解问题的高效实用算法

## 4.0.1 计算几何

## 4.0.2 分布式算法

## 4.0.3 并行算法

## 4.0.4 算法

又穷，确定，有效。

【复杂度】时间，空间

时间复杂度：1,  $\log_2 n$ ,  $n$ ,  $n \log_2 n$ ,  $n^2$ ,  $n^3$ , ...,  $2^n$

### 查找

【顺序查找】平均查找长度： $\frac{n+1}{2}$  time,  $O(N)$

【二分查找】有序排列。比较次数最多  $\lfloor \log_2 n \rfloor + 1$  time,  $O(\log_2 n)$

【散列表】例如，存储空间 10， $p=5$ ，散列函数  $h = key \% p$ ，存储 3,8,12,17,9：线性探测：3,4,2, 5,6 冲突解决：线性探测，伪随机数。

### 排序

稳定、不稳定。【一样的数，保持原顺序，叫稳定】

【插入式：直接插入】新的一个与已经排好的比，插入到位置

【插入式：希尔】数据少时插入排序效率可以。

例如 10 个元素，先  $d=n/2=5$ ，隔 5 个一组，插入排序； $d = d/2 = 2$ ，取奇数是 3；隔 3 个一组，插入； $d = d/2 = 1$ ，全体插入排序。

【选择式：直接选择】每次选剩余最小的。

【选择式：堆排序】完全二叉树。。小顶堆： $k_i \leq k_{2i}, k_i \leq k_{2i+1}$  大顶堆： $k_i \geq k_{2i}, k_i \geq k_{2i+1}$   
所有孩子都更小

从小到大排列：建小顶堆-》取顶-》建小顶堆-》。。。。

例如构造大顶堆：step1，数组顺序构造完全二叉树。step2，最后一个非叶子节点，与其 2 个孩子调整为大顶堆；倒数第 2 个非叶子节点，依次调整。如果有子树，要调整后继续调整子树。

【交换式：冒泡】

【交换式：快速】

【归并排序】

【基数排序】

# Chapter 5 算法综合案例

面向应用的大尺度难解问题的工程实用算法



## Chapter 6 工程算法集成和相应软件 体系结构





## Chapter 7 工程算法分析和评价体系