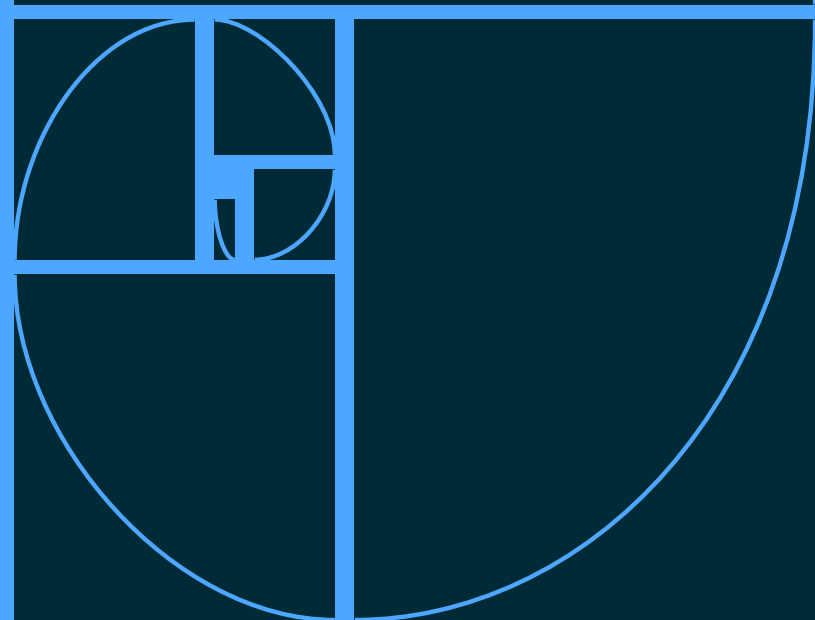
数据结构

2024 年 7 月 8 日



张怀沙（Zesty_Fox）

长郡中学

数据结构，组织、存储、处理数据的工具。

一般与其他领域算法结合使用，如 DP、图论等。

数组

相信大家都非常熟悉，就不多叙述了。

栈

栈：只能从头部删除、增加数据。

```
int st[N],tp=0;
void push(int x) {st[++tp]=x;}
void pop() {--tp;}
```

队列

队列：只能从头部删除数据，尾部增加数据。

一个简单的实现：

```
int q[N],hd=0,tl=0;
void push(int x) {st[tl++]=x;}
void pop() {++hd;}
```

双端队列：头尾均可删除、增加数据，实现类似。

循环队列：节省空间。

STL 中的 queue 提供了普通队列的实现。

STL 中的 deque 提供了循环双端队列的实现，并且支持随机访问。

单调栈

顾名思义，栈里的元素是单调递增/递减的。

本质是排除无效的决策，使得剩下的决策有序，便于寻找最优决策。

单调栈

Luogu P4147 玉蟾宫

给定一个 01 矩阵，找出其中面积最大的、全由 1 组成的矩形。
$N \leq 1000$。

单调队列

同样，队列里的元素是单调递增/递减的。

本质和单调栈相似。

单调队列

Luogu P3572 [POI2014] PTA-Little Bird

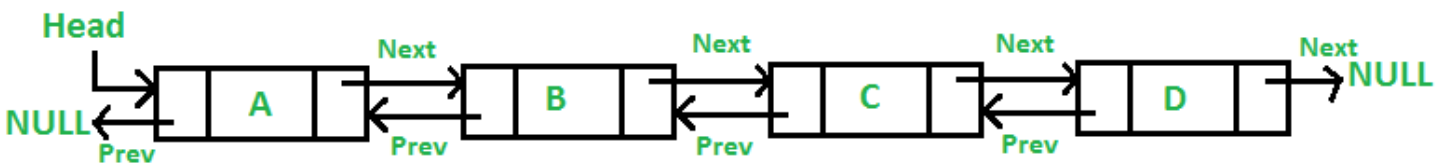
有 n 棵树排成一排，第 i 棵树的高度是 d_i。
有 q 只鸟要从第 1 棵树到第 n 棵树。
当第 i 只鸟在第 j 棵树时，它可以飞到第 $j+1, j+2, \cdots, j+k_i$ 棵树。
如果一只鸟飞到一颗高度大于等于当前树的树，那么它的劳累值会增加 1，否则不会。
最小化每只鸟的劳累值。
$n \leq 10^6, q \leq 25$。

链表

支持 $O(1)$ 在任意位置插入/删除数据，然而随机访问效率为 $O(n)$。有时候会有奇效。

有单向链表与双向链表之分，还有循环链表。

整个链表的结构如图（双向链表）：



STL 中的 `list` 提供了双向链表的实现。

STL 中的 `forward_list` 提供了单向链表的实现。

链表

具体实现可以使用数组 + 下标，能力比较强也可以直接使用指针。后者效率稍高，但调试起来较为困难。

这是一个双向链表的实现。

链表

Luogu P1168 中位数

给定一个整数序列，求每个奇数长度前缀的中位数。
$N \leq 10^6$。

哈希

设想维护一个映射，每一个 key 都对应着唯一的 value，可以通过 key 访问对应的 value。典型例子就是数组，通过下标访问对应的值。

然而，当 key 的范围非常大（如 10^9），甚至不是整数，就无法使用数组了。

此时，需要将 key 转换到一个适当范围内的数，这个转换过程就是哈希函数。

哈希

当 key 是较大的整数（如 10^9），常见的哈希函数是将 key 对一个较小的数取模（并不一定是质数）。

当 key 是字符串，常见的哈希函数是将 key 视作一个 127 进制数并对另一个数取模。

但是，根据生日悖论，多个 key 对应同一哈希值的概率是相当高的，这就会形成哈希冲突，常见的解决办法有拉链法和闭散列法。

如果哈希函数的值域大小为 M，平均情况哈希表查询的时间复杂度为 $O(\frac{N}{M})$，当 $M = \Omega(N)$ 时可视为 $O(1)$。

STL 中的 `unordered_map` 提供了一个哈希表的实现。

哈希

Luogu P3498 [POI2010] KOR-Beads

给定一个整数序列，求当 k 取多少时，将其按顺序切分为长为 k 的子串后子串种类最多。子串可以翻转。

$n \times 2 \times 10^5$。

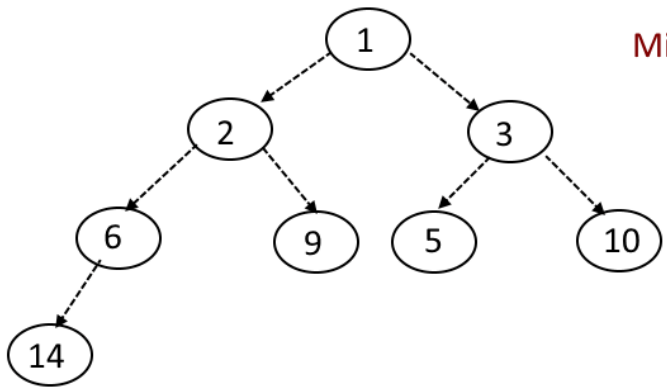
哈希

Luogu P8819 [CSP-S 2022] 星战

给定一个有向图，每次可以加入一条边、删除一条边、禁用一个点所有入边、激活一个点被禁用的所有入边。每次操作后，询问是否有每个点的出度均为 1。

$n, m, q \leq 5 \times 10^5$。

二叉树的存储



Min Heap

0	1	2	3	4	5	6	7	8
	1	2	3	6	9	5	10	14

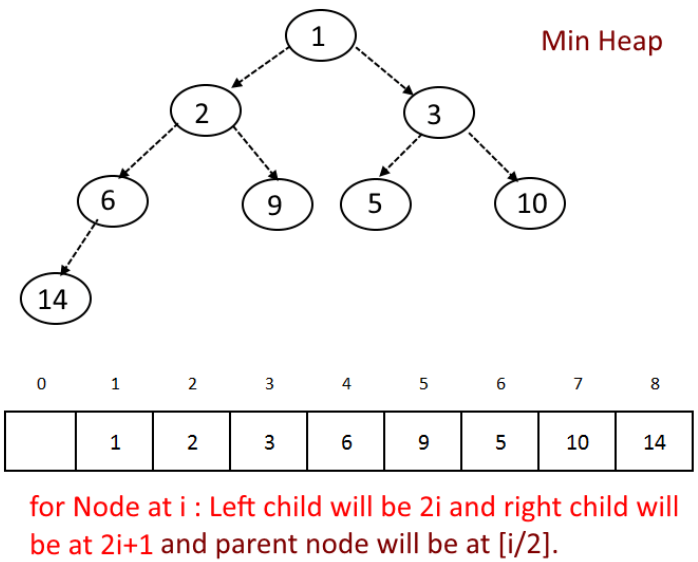
for Node at i : Left child will be 2i and right child will be at 2i+1 and parent node will be at [i/2].

堆

算法竞赛中，最常用的是二叉堆。

以小根堆为例，其支持插入元素、查询当前最小值、删除最小值。

重要性质：父亲的元素值小于儿子。



STL 中的 `priority_queue` 提供了一个大根二叉堆的实现。

堆

一个二叉堆的实现。

堆的拓展

删除任意元素：

堆的拓展

支持查询最大与最小值：

堆

P1090 [NOIP2004 提高组] 合并果子

给定长为 n 的正整数数列，每次合并两个数，花费为两个数之和。
求把所有数合并为一个数的最小代价。
$n \leq 10^6$。

并查集

维护若干个集合，支持合并两个集合、查询某元素所属集合。
通常会使用路径压缩优化，有时也会按秩合并优化。
平均时间复杂度可近似视为常数。

并查集拓展

带权并查集与扩展域并查集：

并查集拓展

维护一个特殊的数据结构：

并查集

Luogu P1640 [SCOI2010] 连续攻击游戏

给定若干二元组，从每个二元组选择一个数，使得选择的所有数的集合中，未出现的最小正整数最大。

$n \leq 10^6, a_i, b_i \leq 10^4$。

并查集

Luogu P2024 [NOI2001] 食物链

动物王国中有三类动物 A,B,C，这三类动物的食物链构成了有趣的环形。A 吃 B，B 吃 C，C 吃 A。
现有 N 个动物，以 $1 \sim N$ 编号。每个动物都是 A,B,C 中的一种，但是我们并不知道它到底是哪一种。
有人用两种说法对这 N 个动物所构成的食物链关系进行描述：X 和 Y 是同类或 X 吃 Y。
此人对 N 个动物，用上述两种说法，一句接一句地说出 K 句话，这 K 句话有的是真的，有的是假的。一句话是假话当且仅当其与前面的真话冲突或自身不合理。你需判断每句话真假。$N, K \leq 10^6$。

以下区间查询默认指区间和。

请注意区间和并不一定是数值运算，实际上更广义。

前缀和

重要的思想，实现区间查询的方式之一。

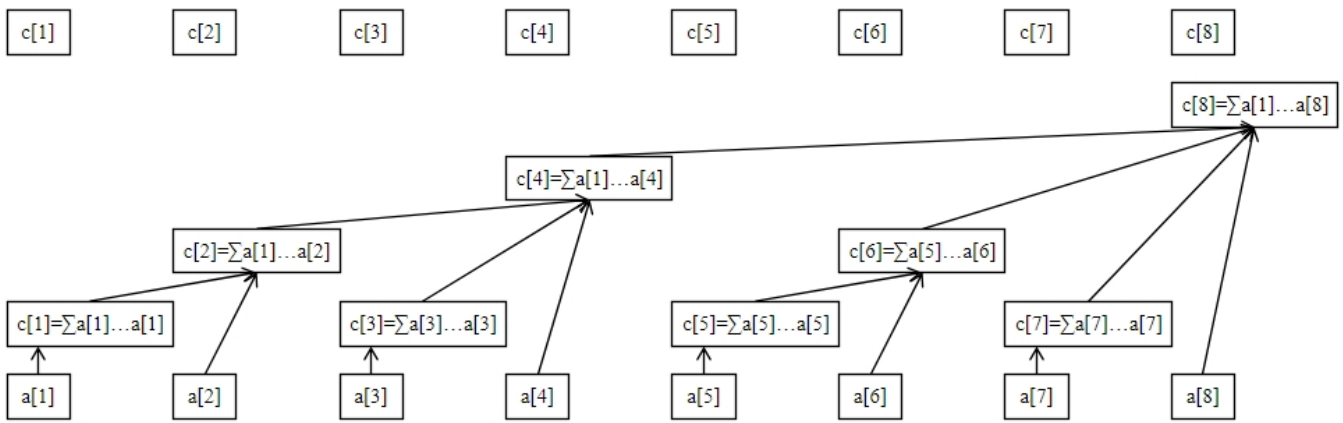
差分

前缀和的逆运算，有时候有奇效。

树状数组

用于维护一个数列，支持单点修改和区间查询。

本质上是二进制拼凑的思想。



效率相当高，常数很小。

树状数组

区间（前缀）查询与单点修改：

树状数组拓展

区间修改，单点查询：差分

树状数组拓展

区间修改，区间查询：两个树状数组

树状数组拓展

倍增查询：支持 $O(\log n)$ 查询一个非负整数序列中，第一个前缀和大于等于指定值的位置。

树状数组

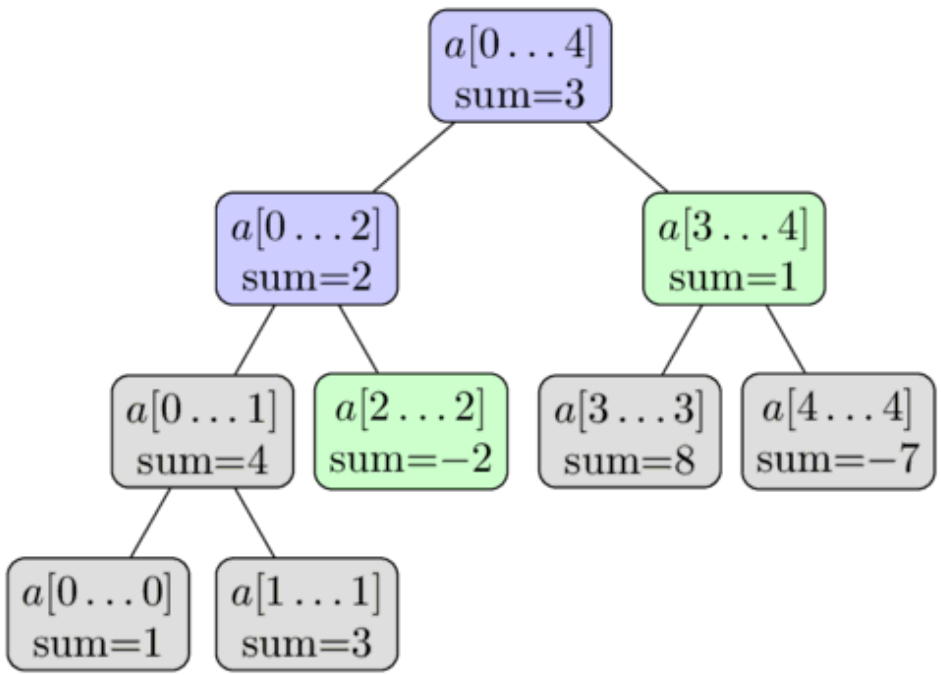
Luogu P6619 [省选联考 2020 A/B 卷] 冰火战士

每位战士有两个属性：温度和能量，有两派战士：冰系战士要求场地温度不低于他的自身温度才能参赛；火系战士要求场地温度不高于他的自身温度才能参赛。
当场地温度确定时，双方能够参赛的战士分别排成一队。冰火双方消耗总能量为双方各自战士能量之和的较小值的两倍。
你需要寻找最佳场地温度：使冰火双方消耗总能量最高的温度的最高值。
有 Q 次操作，每次会动态加入或删除战士，并询问最佳场地温度。
$Q \leq 2 \times 10^6$。

线段树

相当强大，适用性强（只要元素有可加性即可），常数略大。

本质上是分治的思想。



线段树也是二叉树，存储方式同前，空间复杂度为 $O(n)$。

线段树

区间查询与单点修改（上传)：

线段树

重要性质：区间询问只会涉及到 $O(\log n)$ 个结点。

支持区间修改（标记下传）：

线段树拓展

动态开点线段树：

线段树拓展

（带标记的）线段树本质上究竟能维护什么？

- 元素可与标记合并（得到新的元素）
- 元素可合并（得到新的元素，即可加性）
- 标记可合并
- 合并的过程具有结合律

线段树拓展

线段树二分：单点修改，区间查询前缀和大于 S 的第一个位置 p。

线段树

CF115E Linear Kingdom Races

有 n 条连续的从左到右的道路。道路从左到右依次编号为从 1 到 n，因此道路按照升序排列。
所有道路都需要修理，每条路都有与之相关的维修费用，你需要支付这笔费用来修理道路。有若干条件，形如若编号在 $[l_i, r_i]$ 的道路都被修理，你会获得 v_i 元。你的利润被定义为你从比赛中获得的总金额减去你花在修理道路上的钱。
输出你能获得的最大利润。$n, m \leq 2 \times 10^5$。

线段树

Luogu P1712 [NOI2016] 区间

在数轴上有 n 个闭区间从 1 至 n 编号，第 i 个闭区间为 $[l_i, r_i]$ 。
现在要从中选出 m 个区间，使得这 m 个区间共同包含至少一个位置。
换句话说，就是使得存在一个 x ，使得对于每一个被选中的区间 $[l_i, r_i]$，都有 $l_i \leq x \leq r_i$ 。
对于一个合法的选取方案，它的花费为被选中的最长区间长度减去被选中的最短区间长度。
区间 $[l_i, r_i]$ 的长度定义为 $(r_i - l_i)$ ，即等于它的右端点的值减去左端点的值。
求所有合法方案中最小的花费。$n, m \leq 5 \times 10^5$。

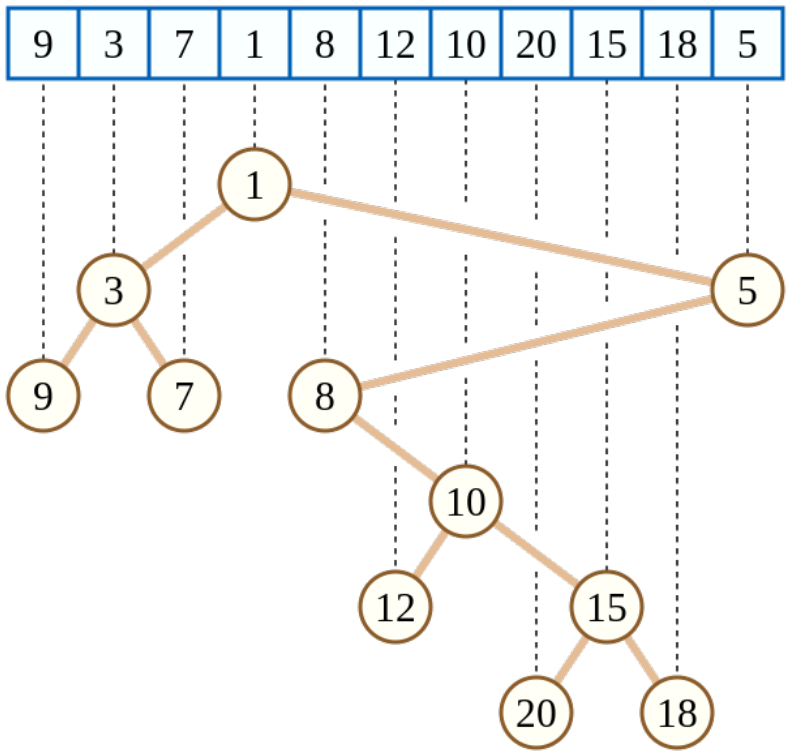
线段树

Luogu P1972 [SDOI2009] HH 的项链

多次询问某区间内数的种类数。

$n, q \leq 10^6$。

笛卡尔树



可使用单调栈 $O(n)$ 建树。

笛卡尔树

假设这颗笛卡尔树由一个连续序列构建，维护最小值。

重要性质：$[l, r]$ 内的区间最小值在 l, r 对应结点的 LCA 上。

笛卡尔树

CF1748E Yet Another Array Counting Problem

对于长度为 n 的序列 x，定义其在子段 $[l;r]$ 的“最左端最大值位置”为最小的满足 $l \le i \le r$ 且 $x_i = \max_{j=l}^{r} x_j$ 的整数 i。

给定整数 n, m 和长度为 n 的序列 a，你需要求出满足下列要求的序列 b 的数量：

- 序列 b 长度为 n，且对任意整数 $i(1 \le i \le n)$ 都有 $1 \le b_i \le m$ 成立。
- 对任意整数 $l, r(1 \le l \le r \le n)$，总有 a, b 在子段 $[l;r]$ 的“最左端最大值位置”相同。

ST 表与 RMQ

RMQ $=$ 区间最值问题。

ST 表：适用静态（没有修改操作）的 RMQ 操作，空间复杂度、初始化复杂度均为 $O(n \log n)$，但单次查询为 $O(1)$。

ST 表与 RMQ

结构与实现：

ST 表与 RMQ

除了 ST 表，前面所说线段树也可实现 RMQ，但树状数组不行（为什么?)。

RMQ 还有离线单调栈算法（空间复杂度较优)，甚至有 $O(n)$ 初始化、查询 $O(1)$ 算法。

平衡树

在这个层次，基本不会考自己编写平衡树，而以应用为主。

以 STL 中的 `map` 为例（内部实现是平衡树），如果 key 为整数，其支持以 $O(\log n)$ 的时间插入、删除任意元素、查询某个元素前驱后继。

STL 中的数据结构

实际上，很多数据结构并不需要自己编写，STL 提供了现成的实现。

另外，对于某些数据结构，pb_ds 库提供了一部分比 STL 更强大的实现。

具体用法建议参考网上文档（如 OI-wiki 的 STL 介绍、pb_ds 介绍），在此不多赘述。

完结撒花！