CSP-S 数据结构提升与应用

SRwudi

October 6, 2019

开篇

什么是数据结构？

数据结构是计算机存储、组织数据的方式。
抽象数据结构的具体实现。
通过数据结构的存储和应用，可以提高计算机运作的效率。

开篇

什么是数据结构？

数据结构是计算机存储、组织数据的方式。
抽象数据结构的具体实现。
通过数据结构的存储和应用，可以提高计算机运作的效率。

在 OI 的运用当中，数据结构题通常表现什么方面的性质。

大规模输入。
有修改和查询操作。
或者是一些隐藏的不容易看出来的数据结构题需要你去维护。

开篇

本次讲课主要涉及 CSP-S 范围内的各种类型的数据结构。

基础知识和具体实现以复习和理解为主，主要主攻的应用题怎么做。

数据结构的具体实现或者代表性的基础题目不会过多赘述，本篇主讲例题都是省选难度 -NOI 难度且具有一定思维挑战的题目。

二叉堆

二叉堆

维护一列数据中的最小值/最大值
加入：$O(\log n)$
删除：$O(\log n)$
查询：$O(1)$
实现：stl/pbds

tips: 前 k 大，最大最小。

如何使用 stl 的 priority_queue 实现方便的删除。

如何使用 stl 的 priority_queue 实现方便的删除。

Tips: 再开一个堆表示删除堆，存储我要删除的元素，然后每当原堆顶元素和当前删除堆顶元素相同后，同时删除。

中位数

维护一个数的集合：
操作 1：插入一个数 x
操作 2：询问这个集合的中位数。

序列合并

有两个长度都是 N 的序列 A, B，在 A 和 B 中各取一个数相加可以得到 N^2 个和，

求其中最小的 M 个这样的两个数的和

$N, M <= 100000$

镇静剂

有一天，病房里来了 n 位紧急需要紧急救治的病人，而你手上有 V 毫升镇静剂。

对于第 i 位病人，他需要治疗的紧急程度为 a_i，让他镇静下来所需的剂量为 b_i。

你可以选择一些病人，把你现在手头的镇静剂给他们注射，让他们安静下来。

如果你选择了 k 个病人：$c_1, c_2, c_3, \ldots, c_k$，那么第 j 个病人 c_j 会得到

$$\frac{a_{c_j}}{\sum_{i=1}^{k} a_{c_i} \times V}.$$

毫升镇静剂，如果这个值不小于 b_{c_j}，那么他就会镇静下来。

NOI2010 超级钢琴

题目大意

给一段数有正有负的序列，数 L, R，求所有前 K 大长度介于 $[L, R]$ 的区间的和。

$N, K \leq 500000$

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l，右端点在 $[r_1, r_2]$ 之内的区间总和。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l, 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l, 右端点在 $[r_1, r_2]$ 之内的的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l, 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l, 右端点在 $[r_1, r_2]$ 之内的的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

核心思路（贪心 + 扩展）：每次取出一个堆顶的元素 (l, r_1, r_2)，用其最大和区间更新答案，并弹出该元素。最后加入两个新元素 $(l, r_1, r - 1), (l, r + 1, r_2)$. 做 K 次结束。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l, 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l, 右端点在 $[r_1, r_2]$ 之内的的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

核心思路（贪心 + 扩展）：每次取出一个堆顶的元素 (l, r_1, r_2)，用其最大和区间更新答案，并弹出该元素。最后加入两个新元素 $(l, r_1, r-1), (l, r+1, r_2)$. 做 K 次结束。

如何寻找最大和区间：前缀最大和 ST 表。

NOI2010 超级钢琴

用三元组 (l, r_1, r_2) 表示代表左端点为 l, 右端点在 $[r_1, r_2]$ 之内的区间总和。

以左端点为 l, 右端点在 $[r_1, r_2]$ 之内的的最大和区间 $[l, r]$ 作为 (l, r_1, r_2) 的键值维护堆。

核心思路（贪心 + 扩展）：每次取出一个堆顶的元素 (l, r_1, r_2)，用其最大和区间更新答案，并弹出该元素。最后加入两个新元素 $(l, r_1, r-1), (l, r+1, r_2)$. 做 K 次结束。

如何寻找最大和区间：前缀最大和 ST 表。

堆的初始值：所有的 $(i, i+L-1, i+R-1)$。

二叉堆

维护一张图中两个点的连通性
支持两个操作：
加边：$O(\log n)$
查询两个点的连通性：$O(\log n)$

路径压缩: $O(\log n)$
按秩合并：$O(\log n)$
两个都用: $O(\alpha n)$

可撤销并查集

加边，撤销上次加边，维护连通性

就单独按秩合并就好，然后再开一个栈表示之前做过的操作。

例题 1

支持两个操作：
1. 加边
2. 询问 $\sigma_{i=1} n \sigma_{j=i+1}^{n} L(i,j)$.

$L(i,j)$ 表示两个点最早什么时候连通。不连通为 0。

例题 2

一开始加一些边。最后给若干个询问，求两个点是在什么时候连通的。

例题 3

给一个数组 $a[1...n]$，开始全是 0，每次有两种操作：
令 $a[x]=1$。
输出 $a[x...n]$ 中第一个 0 的位置。

Exclusive-OR

有 n 个 $0/1$ 变量，每次告诉你 $x_i = 0/1$ 或者告诉你 $x_i xor x_j = 1$ 这么一个关系，最后询问你一个数等于多少，或两个数异或值。

2-set

有 n 个互不相同的数，你需要将这 n 个数分成两个集合 A 和 B，使得：

- 如果 x 在 A 的话，那么 $a - x$ 也要在 A。
- 如果 x 在 B 的话，那么 $b - x$ 也要在 B。

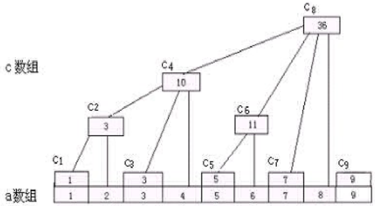
相同区间

有 n 个序列 $a[1...N]$，其中 $0 <= a[i] <= K$，并且告诉你 m 条

信息：$a[l1...r1] = a[l2...r2]$，求有几种满足条件的序列。

简介

可以方便的维护一类前缀和，每个操作复杂度 $O(\log n)$。



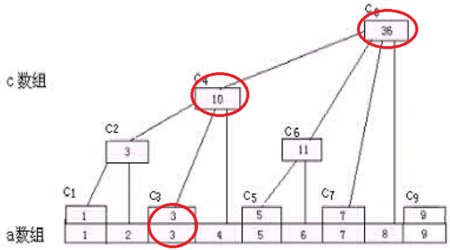
$C_1 = a_1$
$C_2 = a_1 + a_2$
$C_3 = a_3$
$C_4 = a_1 + a_2 + a_3 + a_4$
$C_5 = a_5$ ....
$C_8 = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8$ ....

求和

向下每次减去 $lowbit(x)$ 求和

```
int getsum(int x) {
    int ans = 0;
    while (x) ans += c[x], x -= lowbit(x);
    return ans;
}
```

修改



向上每次加上 $lowbit(x)$ 更新。

```
int update(int x, int d) {
    while (x <= n) c[x] += d, x += lowbit(x);
}
```

应用

维护一个数的序列。需要支持：

应用

维护一个数的序列。需要支持：
Level1：单点加减，区间求和。

应用

维护一个数的序列。需要支持：
Level1: 单点加减，区间求和。
Level2：区间加减，单点询问。

应用

维护一个数的序列。需要支持：
Level1: 单点加减，区间求和。
Level2：区间加减，单点询问。
维护差分数组 $b_i = a_i - a_{i-1}$。则 $a_i = \sigma_{j=1}^{i} b_j$。

Level3：区间加减，区间求和。

Level3：区间加减，区间求和。
维护差分数组 $b_i = a_i - a_{i-1}$，和另一个数组 $i * b_i$。则前缀和

$$S_n = \sum_{i=1}^{n} \sum_{j=1}^{i} b_j$$
$$= \sum_{j=1}^{n} \sum_{i=j}^{n}) b_j$$
$$= \sum_{j=1}^{n} (n - j + 1) b_j$$
$$= \sum_{i=1}^{n} (n - i + 1) b_i$$
$$= (n+1) \sum^{n} b_i - \sum^{n} ib_i$$

二维树状数组

维护一个数的序列。需要支持每个操作 $O(\log^2 n)$：

二维树状数组

维护一个数的序列。需要支持每个操作 $O(\log^2 n)$：
Level1：单点加减，矩阵求和。

二维树状数组

维护一个数的序列。需要支持每个操作 $O(\log^2 n)$：
Level1: 单点加减，矩阵求和。
Level2：矩阵加减，单点询问。

二维树状数组

维护一个数的序列。需要支持每个操作 $O(\log^2 n)$：
Level1：单点加减，矩阵求和。
Level2：矩阵加减，单点询问。
Level3：矩阵加减，矩阵求和。

PKU2352Star-二维偏序

二维平面坐标第一维象限中分布着 n 个点，询问对于每个点，两个维度均小于该点的点的个数有多少个。

$1 \le N \le 15000, 0 \le X, Y \le 32000$

最长回文子序列

给你一个长度为 N 的序列 A，满足每个数字最多出现 K 次，求最长回文子序列的长度。

例：1,2,7,2,1 是 1,3,2,7,4,2,1 的一个最长回文子序列。

$N \leq 10^5, K \leq 4.$

CodeForces - 830B

给你一串数字，操作过程如下：不断把队首数字掉到队尾，如果当前队首数为队列中的最小值 (注意可能会有多个最小值)，那么就删除队首。两种操作都会使操作数加一。问你至少要操作多少次才能将队列删空。

2019 Multi-University Training Contest 2

给一个序列，在满足单调递增或者单调递减或者先增后减的最长子序列集合里找到下标字典序最大以及最小的两个子序列，输出这两个子序列里元素的下标

TJOI2017 异或和

给定一个序列 $a_1, a_2, ..., a_n$，满足 $a_1 + a_2 ... + a_n \leq 10^6$，输出输出这个序列所有的连续和的异或值。

线段树

线段树（常规）

维护可区间合并值。例如最大值最小值，区间之和，最大子区间和，连续颜色段数量等等。

可维护的修改操作只要能支持懒标记合并即可。比如乘法，加法，赋值，01 取反。

支持 $O(\log n)$ 区间修改和区间查询。

基础例题

给定一个数组，支持两如下操作：

- 对一个区间的每个位置加上一个数，减掉一个数，乘上一个数
- 询问一段区间的和，最大值，最小值

Dance

给定一个 01 串，支持两种操作：

- 查询一段区间内 01 交替的序列的数量
- 修改一个位置的 0 或 1

HDU 3308 LCIS

提交地址：
http://acm.hdu.edu.cn/showproblem.php?pid=3308
给出一个的数列，支持两种操作：

- 将第 a 个数替换为 b (下标从 0 开始)
- 输出区间 $[l, r]$ 的最长连续递增子串。

Vijos 1883 月光的魔法

提交地址：https://vijos.org/p/1883

给定 n 个圆心在 x 轴上的圆，任意两个圆不相交不重合。圆心都在整点上，所有圆的半径都是整数。求这些圆把这个平面分割成了多少块。

BZOJ3211 花神游历各国

题目大意

给一个长度为 N 的序列，需要支持以下两个操作:

- 对于一段区间，每个值 sqrt 下取整操作
- 区间求和

$$N \leq 100000, M \leq 200000, data_i \leq 10^9$$

UOJ 222 NOI2016 区间

提交地址：http://uoj.ac/problem/222

在数轴上有 n 个闭区间 $[l_1, r_1], [l_2, r_2], \ldots, [l_n, r_n]$。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 $l_i \le x \le r_i$。

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。如果不存在合法的方案，输出 -1。

JSOI2014 奇怪的计算器

题目大意

有 N 个数，一共会对这 N 个数执行 M 个指令（对没个数执行的指令都一样），每一条指令可以是以下四种指令之一：(这里 a 表示一个正整数)

1：加上 a

2：减去 a

3：乘以 a

4：加上 a*X（X 是数最开始的初值）

该计算器有个奇怪的特点。每进行一个指令，若结果大于 R 则变成 R，同理若结果小于 L，则变成 L。求这 N 个数最后的结果。

$N, M \leq 200000$

等差区间加强版

题目大意

给出一个长度为 N 的序列，需要支持以下两个操作：

- 单点修改
- 询问每个区间内数字升序排列后是否为等差数列.

$N \leq 100000, M \leq 200000, data_i \leq 10^9$

等差区间加强版

若想知道区间内数是否能够成等差数列，显然要求出区间内的最值（即数列首项和末项），则即可算出对应的公差。

等差区间加强版

若想知道区间内数是否能够成等差数列，显然要求出区间内的最值（即数列首项和末项），则即可算出对应的公差。

注意到若区间内数能成等差数列，则需满足下列中的一项：

1. 公差为零；

2. 公差不为零，则数列内无相同元素，那么公差为相邻的数值差的最大公因数。

等差区间加强版

若想知道区间内数是否能够成等差数列，显然要求出区间内的最值（即数列首项和末项），则即可算出对应的公差。

注意到若区间内数能成等差数列，则需满足下列中的一项：

1. 公差为零；

2. 公差不为零，则数列内无相同元素，那么公差为相邻的数值差的最大公因数。

于是需要维护，区间最大值最小值，差分数列的区间 gcd，以及每个数前驱相同元素的位置。

等差区间加强版

若想知道区间内数是否能够成等差数列，显然要求出区间内的最值（即数列首项和末项），则即可算出对应的公差。

注意到若区间内数能成等差数列，则需满足下列中的一项：

1. 公差为零；
2. 公差不为零，则数列内无相同元素，那么公差为相邻的数值差的最大公因数。

于是需要维护，区间最大值最小值，差分数列的区间 gcd，以及每个数前驱相同元素的位置。

维护每个数前驱相同元素的位置可以用 map 套 set。

单调上升序列

题目大意

给一个长度为 n 的数列，每次修改其中一个数，每次询问一个区间的单调上升序列长度。

$1 \leq n \leq 10^5$

单调上升序列就是指在序列中权值比之前的数都大的数的个数。

单调上升序列

考虑维护线段树。

单调上升序列

考虑维护线段树。
对于节点 x，记录两个权值 Len_x 和 Mx_x

单调上升序列

考虑维护线段树。

对于节点 x，记录两个权值 Len_x 和 Mx_x

定义函数 $get(x, M)$ 表示对于节点 x 所代表的区间 $[l_x, r_x]$ 中最长上升序列 $> M$ 的部分。

单调上升序列

考虑维护线段树。

对于节点 x，记录两个权值 Len_x 和 Mx_x

定义函数 $get(x, M)$ 表示对于节点 x 所代表的区间 $[l_x, r_x]$ 中最长上升序列 $> M$ 的部分。

那么我们有更新操作 $Len_x = Len_{lc_x} + get(rc_x, Mx_{lc_x})$

单调上升序列

考虑维护线段树。

对于节点 x，记录两个权值 Len_x 和 Mx_x

定义函数 $get(x, M)$ 表示对于节点 x 所代表的区间 $[l_x, r_x]$ 中最长上升序列 $> M$ 的部分。

那么我们有更新操作 $Len_x = Len_{lc_x} + get(rc_x, Mx_{lc_x})$

对于函数 $get(x, M)$ 我们有 $O(\log n)$ 时间内递归求解做法：

单调上升序列

考虑维护线段树。

对于节点 x，记录两个权值 Len_x 和 Mx_x

定义函数 $get(x, M)$ 表示对于节点 x 所代表的区间 $[l_x, r_x]$ 中最长上升序列 $> M$ 的部分。

那么我们有更新操作 $Len_x = Len_{lc_x} + get(rc_x, Mx_{lc_x})$

对于函数 $get(x, M)$ 我们有 $O(\log n)$ 时间内递归求解做法：

递归求解

$$get(x, M) = get(lc_x, M) + Len_x - Len_{lc_x} \quad (mx_{lc_x} > M)$$

$$get(x, M) = get(rc_x, M) \quad (mx_{lc_x} \leq M)$$

单调上升序列

考虑维护线段树。

对于节点 x，记录两个权值 Len_x 和 Mx_x

定义函数 $get(x, M)$ 表示对于节点 x 所代表的区间 $[l_x, r_x]$ 中最长上升序列 $> M$ 的部分。

那么我们有更新操作 $Len_x = Len_{lc_x} + get(rc_x, Mx_{lc_x})$

对于函数 $get(x, M)$ 我们有 $O(\log n)$ 时间内递归求解做法：

递归求解

$$get(x, M) = get(lc_x, M) + Len_x - Len_{lc_x} \quad (mx_{lc_x} > M)$$

$$get(x, M) = get(rc_x, M) \quad (mx_{lc_x} \leq M)$$

时间复杂度？

单调上升序列

考虑维护线段树。

对于节点 x，记录两个权值 Len_x 和 Mx_x

定义函数 $get(x, M)$ 表示对于节点 x 所代表的区间 $[l_x, r_x]$ 中最长上升序列 $> M$ 的部分。

那么我们有更新操作 $Len_x = Len_{lc_x} + get(rc_x, Mx_{lc_x})$

对于函数 $get(x, M)$ 我们有 $O(\log n)$ 时间内递归求解做法：

递归求解

$$get(x, M) = get(lc_x, M) + Len_x - Len_{lc_x} \quad (mx_{lc_x} > M)$$

$$get(x, M) = get(rc_x, M) \quad (mx_{lc_x} \le M)$$

时间复杂度？

询问和修改操作均为 $O(\log n^2)$。

CTSC 模拟栈 By: 吉如一

题目大意

你有 n 个单调栈排成一排，你要支持对于连续一段区间内的单调栈压入同一个数 x，并支持询问某一个单调栈内的数的总和。

$1 \leq n, m \leq 2 * 10^5$

CTSC 模拟栈 By: 吉如一

考虑离线。

CTSC 模拟栈 By: 吉如一

考虑离线。
以时间为轴建立线段树。

CTSC 模拟栈 By: 吉如一

考虑离线。

以时间为轴建立线段树。

将栈的序列轴改成时间轴，那对于一段连续的区间内压入一个数，就等价于在新的时间轴上变成添加和删除操作。

CTSC 模拟栈 By: 吉如一

考虑离线。

以时间为轴建立线段树。

将栈的序列轴改成时间轴，那对于一段连续的区间内压入一个数，就等价于在新的时间轴上变成添加和删除操作。

求某个时间内的单调栈数总和，变成询问在某个时间点上的某个区间的最长上升序列。

CTSC 模拟栈 By: 吉如一

考虑离线。

以时间为轴建立线段树。

将栈的序列轴改成时间轴，那对于一段连续的区间内压入一个数，就等价于在新的时间轴上变成添加和删除操作。

求某个时间内的单调栈数总和，变成询问在某个时间点上的某个区间的最长上升序列。

问题等价于支持修改的在线区间最长上升序列。