

# 珂朵莉树

📅 2022-08-03 | 📅 2022-08-04 | 📁 OI & ACM | 👁 3

📄 2k | ⌚ 2 分钟

## 简介

其实珂朵莉树并不是一种树，我觉得应该算作一种处理颜色段的方法

## 模板

`auto` 的类型推断不是 `c++11` 的语法，小心 `ce`

另外珂朵莉树如果存在需要遍历区间内所有颜色段且遍历后不合并的情况下复杂度是不对的，除非数据随机

```
1  struct Chtolly {
2      int l, r;
3      mutable int v; // 需要修改的值需要加 mutable 关键字
4
5      Chtolly(int l = 0, int r = 0, int v = 0) : l(l), r(r), v(v) {}
6      friend bool operator < (const Chtolly &u, const Chtolly &v) { return u.l < v.l
7  }; set<Chtolly> S;
8
9  auto find(int k) { return prev(S.upper_bound(Chtolly(k))); } // 返回 k 所在的颜色段
10
11 auto split(int k) { // 将 k 所在的颜色段 [l, r] 分割成 [l, k - 1] 和 [k, r] 同时返回
12     auto it = S.lower_bound(Chtolly(k));
13     if (it != S.end() && it -> l == k) return it;
14     it = prev(it); Chtolly t = *it;
15     if (it -> r < k) return S.end();
16     return S.erase(it), S.emplace(t.l, k - 1, t.v), S.emplace(k, t.r, t.v).first;
17 }
18
19 void assign(int l, int r, int v) { // 将 [l, r] 的颜色变成 v
20     auto itr = split(r + 1), itl = split(l);
21     S.erase(itl, itr);
```

```

22     S.emplace(l, r, v);
23 }
24
25 auto merge(int k) { // 将 k 所在的颜色段和与其相邻的同色颜色段合并成一个颜色段
26     auto it = find(k), itl = it, itr = it;
27     while (itr != S.end() && itr -> v == it -> v) ++itr;
28     while (itl != S.begin() && prev(itl) -> v == it -> v) itl = prev(itl);
29     Chtolly t = Chtolly(itl -> l, prev(itr) -> r, it -> v);
30     return S.erase(itl, itr), S.insert(t).first;
31 }

```

## 例题

1. 简要题意：给定一个长度为  $n$  的序列  $a_i$ ，现在有  $m$  个操作，操作有四种，第一种操作给定区间  $[l, r]$  和权值  $v$ ，将  $[l, r]$  所有数加上  $v$ ；第二种操作给定区间  $[l, r]$  和权值  $v$ ，将  $[l, r]$  内所有数变成  $v$ ；第三种操作给定区间  $[l, r]$  和  $k$ ，求  $[l, r]$  内第  $k$  小的数；第四种操作给定区间  $[l, r]$  和  $x, y$ ，求  $(\sum_{i=1}^n a_i^k) \bmod y$

$n, m \leq 10^5$ ，数据随机

简要题解：因为数据随机，所以我们直接使用珂朵莉树维护权值相同的段即可，据说随机数据情况下，只有  $O(\log \log n)$  段

CF 896C Willem, Chtholly and Seniorious

2. 简要题意：现在有一个长度为  $n$  的序列，每个位置有权值  $a_i$  和颜色  $c_i$  两个属性，现在有  $m$  次操作，操作有四种，第一种操作给定  $x$  和  $c$ ，表示将与  $x$  最近的  $c$  个数(包括  $x$  自身在内)的颜色都变成一种未出现过的颜色；第二种操作给定  $x$  和  $y$ ，将与  $y$  颜色相同且与  $y$  相连的颜色段的颜色都变成  $x$  所在颜色段的颜色；第三种操作给定  $x$  和  $v$ ，将与  $x$  同色的所有位置的权值都加  $v$ ；第四种操作给定  $x$ ，求  $x$  的权值

$n \leq 10^8, m \leq 10^5$ ，强制在线

简要题解：我们可以用珂朵莉树维护颜色段，因为我们所有的操作在遍历到一个颜色段后都会将其删除，所以复杂度是正确的

对于权值修改，我们直接在全集记录每种颜色的权值增量，这样我们只需要在一个位置的颜色改变的时候做区间修改即可，注意到  $n$  很大，我们可以使用动态开点线段树或树状数组，动态开点树状数组需要 *map* 或者手写 *hash* 表在常数上表现不如动态开点线段树，时间复杂度  $O(m \log n)$

## 2022杭电多校5 A Pandaemonium Asphodelos: The First Circle (Savage)

-----

本文结束  感谢阅读

-----

# Tech # 珂朵莉树

---

< 2022牛客多校4 E Jobs (Hard Version)

CF 896C Willem, Chtholly and Seniorious >

© 2020 – 2022  DDOSvoid

 1.8m |  27:07

9089 |  17834

由 Hexo & NexT.Gemini 强力驱动