Data Structures

数据结构

OYJason

THU.IIIS

2023.12

你应该已经会了 (模板)

- 并查集
- 单调栈/队列
- 线段树
- 树状数组
- splay

可并堆

- 左偏树
  - 设 $d[x]$ 为 x 一直向右儿子走的距离
  - 取 $d[son]$ 更小的放在右边
  - 合并 (x, y) 时 (小根堆):
    - 设 x key 值更小
    - 递归合并 x 的右儿子和 y
    - 看情况交换 x 左儿子的新右儿子

线段树

- 什么，你不会线段树？
  - 线段树分治/整体二分
  - 线段树合并
  - 可持久化线段树
  - Segment Beats
  - ...

树状数组

- 常数小 + 代码好写
- 高维情况

$$s(x_1, x_2, \cdots, x_k) = \sum_{i_1=L(x_1)}^{x_1} \sum_{i_2=L(x_2)}^{x_2} \cdots \sum_{i_k=L(x_k)}^{x_k} a(i_1, i_2, \cdots, i_k)$$

- 其中 $L(x) = x - lowbit(x) + 1$

LCT

" 动态树剖"

- 每条 splay 按照 dfs 序维护一条“重”链
- 动态
  - 换根
  - 加/删边
- $access(x)$ 把 x 到根的路径设为重链，且 x 为链底
- 求 $x-y$ 路径信息：$makeroot(x), access(y)$

Assumption

- 默认上述数据结构已经掌握

一些花哨数据结构

- 树套树
  - CTSC2008 网络管理
- KD-Tree
  - BZOJ4066 简单题
- 李超树
  - HEOI2013 Segment
- Kruskal 重构树
  - NOI2018 归程
  - BZOJ3551 Peaks 加强版
- 全局平衡二叉树
  - 洛谷 P4751 动态 DP 加强版

李超树

- 插入线段
- 求与 $x = k$ 相截线段中 y 最大坐标

- 两种操作
- 在 (x_1, y_1) 到 (x_2, y_2) 插入线段
- 求所有线段与 $x = k$ 相截的 y 最大值
- 输入均为整数

Sol

- 线段树，先考虑直线
  - 每个节点存与区间中点相截的最高线段
  - 对于其余线段，左右一侧有被完爆
  - 递归插入处理另一侧
- 直线 $\rightarrow$ 线段，拆成线段树上区间
- 查询：线段树从下往上扫取最大值
- $O(n\log^2 n)$

Kruskal 重构树

- 并查集，但是保留合并历史
- 第 i 小的树边 (u, v) 新建节点 $n+i$，左右儿子分别为 u，v 当前所在的二叉树的根
- 以 $n+i$ 为根的子树恰好为刚刚连出前 i 条边时第 i 条边所在的连通块

- 给定一个无向图，每条边有 l, a 两个属性
- Q 次询问，每次问给定 v, p，从 v 出发，一开始可以开车免费通过任意 $a \geq p$ 的边，之后必须徒步花费 l 属性作为边长走到 1 号点，询问最短路径长度
- $n, m, Q \leq 10^5$ 强制在线

Sol

- 设 x 到 1 以 l 为边权的最短路为 $d(x)$
- 转化为求一个“前缀”最大生成树的连通块的 $d(x)$ 最小值

Sol

- 以 a 为边权建出 Kruskal 重构树
- 每个节点存储子树 $d(x)$ 最小值
- 倍增找到重构树上 $a \geq p$ 的最高祖先
- $O(m \log m + Q \log n)$

全局平衡二叉树

- 静态 LCT
- 每个点到根距离 $O(\log)$

动态 DP 加强版

- 单点修改，查询树上最大独立集
- 要求 $O(n \log n)$

Sol

树链剖分，维护链矩阵连乘积

- 树剖 + 线段树 $O(\log^2)$
- 树剖 + 平衡树 $O(\log)$

构建平衡树时取左右最大值最小的点为根，左右递归

Assumption

- 热身题
- 若无特殊说明，默认 $n, m, q \le 10^5$

- 维护一个森林，支持
  - 加入一条边，保证还是森林
  - 取一条边，求经过这条边的简单路径数量

Sol

经过这条边的简单路径数量 = 删去这条边两边树大小的乘积

- LCT 维护子树大小
  - 每个点维护虚字树大小的和
  - 轻重链切换时注意维护

给一个连通图，每条边有一个 len 位的二进制数
支持三种操作:

- 加一条新边
- 删一条新边
- 修改一条新边

每次操作后求最大的从 1 出发回到 1 的路径异或值
$n, m \leq 500 \quad q, len \leq 1000$

Sol

注意到连通，如果无删除，线性基即可
路线可以分解为若干次:

- 用旧边造一棵生成树，考虑所有新边和多余的旧边
- 沿着树边走 $\to$ 走一条树边 $\to$ 沿着树边走回去
- 把这样的路径异或和丢到线性基中

Sol

修改 = 删除 + 加边

- 线段树分治，以操作时间为下标
- 把新边按照存在时间的区间加入到线段树节点中
- dfs 线段树
  - 在当前节点插入对应的边所有边
  - 离开时撤销
- $O((m + Q\log Q)\frac{len^2}{32})$，实际上很难卡满

BJOI2019 删数

- 给一个序列，进行删数
  - 设当前剩余长度为 k
  - 删去所有恰好为 k 的数
  - 重复以上两步，直到序列空或不能删为止
- 支持单点修改，整体 ± 1
- 每次操作后求是否能够删完

Sol

依次递进的结论

- 顺序无所谓，只有 $[1, n]$ 的数有意义
- 必须要有 n，次大出现的数为 $n - n$ 出现次数
- 递归到子问题

Sol

- 设 i 出现了 cnt_i 个
- $[i - cnt_i + 1, i]$ 恰好覆盖 $[1, n]$
- 未覆盖的总长恰好为需要修改的数量
  - 每次找到最小的被覆盖不止一次的位置
  - 找到覆盖它的最小的 i，删掉一个

Sol

考虑修改

- 线段树维护每个位置被覆盖次数
- 记录区间最小值，区间最小值个数
- 整体 ± 1, 平移需要覆盖的区间
- 注意到忽略所有 $i > n$ 的贡献
- 整体 ± 1 时单独考虑 $i = n, n+1$ 的情况

Assumption

- 难度题
- 若无特殊说明，默认 $n, m, q \leq 10^5$

- 有 n 个变量和 n 个方程，对于 $1 \le i \le n$
- $x_i \equiv k_i \times x_{p_i} + b_i \mod 10007$
- 支持操作：
  - 询问 x_i 的解，判断是否无解/多解/求唯一解
  - 修改某个方程 (k_i, p_i, b_i)

Sol

连接 x_i 到 p_i，注意到一定形成基环森林

- 先随便断开环上的一条边
- 根的值 x_r 确定所有值
- 加上断开的边形成 $x_r \equiv kx_r + b$
  - 无解 $k = 1, b \neq 0$
  - 多解 $k = 1, b = 0$
  - 否则 $x_r = b(1-k)^{-1}$

Sol

LCT 维护路径上方程的复合

- 修改 i 时若在环上，先把 i 转到根
- 询问时 access
- 注意判断路径上是否出现乘 0 的情况

HNOI2017 影魔

给定一个排列 a，定义点对 $(i,j), i<j$ 的贡献为

- 若 $\max_{i<k<j} a_k < \min(a_i, a_j)$，为 p_1
- 若 $\min(a_i, a_j) < \max_{i<k<j} a_k < \max(a_i, a_j)$，为 p_2

q 次询问 $[l, r]$ 内点对的贡献和

Sol

对于任意点对 $(i,j), i<j$，取 $i<k<j$ 且 a_k 最大的 k
考虑哪些 (i,j) 点对会对应这个 k

- 设 s_k, t_k 分别为两侧最靠近 k 且大于 a_k 的位置
- (s_k, t_k) 构成第一类 (i,j)
- $s_k < i < k, j = t_k$ 构成第二类 (i,j)
- $i = s_k, k < j < t_k$ 构成第二类 (i,j)

Sol

通过单调栈求 s_k, t_k，问题转化为

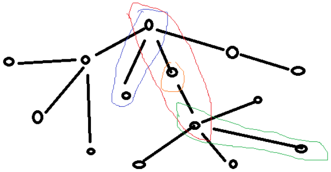
- 对 $x = t$ 直线上的一段区间加
- 对 $y = s$ 直线上的一段区间加
- 单点加
- 求 $[l, r] \times [l, r]$ 的和

注意到：询问时 x, y 故可将 $y = s$ 转化成 $x = t$ 加以 x 为版本下标建可持久化线段树，标记永久化即可

SDOI2017 树点涂色

- 定义路径权值：连续颜色段的数量
- 维护一棵树，支持如下三种操作：
  - 将 x 到根路径上每一个点都修改成新的颜色
  - 查询某条路径权值
  - 查询某颗子树内点到 1 路径最大权值
- $n, Q \leq 10^5$

- 设 $D(x)$ 为 x 到根路径的点权
- 则第二种操作等价于求 $D(u) + D(v) - 2D(lca) + 1$，只需维护 D 即可
- 修改相当于 Link-Cut-Tree access，过程中：
  - 连接到重链等价于子树 $D - 1$
  - 从重链中断开相当于子树 $D + 1$
- 用线段树维护即可，$O(n \log^2 n)$



BZOJ4695 最假女选手

- 维护一个序列 a，支持:
  - 给定 u, 序列区间对 u 取 $\max/\min$
  - 给定 v 区间 $+v$
  - 求区间最值
  - 求区间和

Sol

难点在于区间取 min / max, 假设只取 min.

- 线段树维护区间最小值 mx 和严格次小值 se
- 若取最大值的数介于 mx 和 se 之间只需要改 mx
- 否则暴力递归

取 max 同理

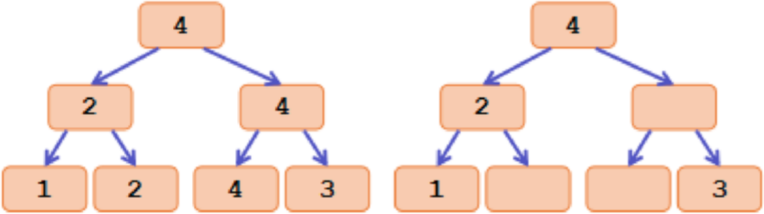
注意 pushdown 标记合并的细节

- $+= x, \min(a) \Rightarrow \min(a - x), += x$
- $+= x, \max(a) \Rightarrow \max(a - x), += x$

Sol

复杂度分析

- 给线段树节点打上标记 $t_x = mx_x$
- 删除所有与父结点标记相同的标记



注意到此时 se 即为子树内最大的标记

设 $T = \sum_{t_x} depth(x)$，即所有标记的深度和
假设对 v 取 min，考虑对标记总数的影响
每次非暴力递归会使 T 增加 $O(\log^2)$
初始时 $T = O(n \log n)$

Sol

- 暴力递归相当于消除子树内所有 $t_x \geq v$ 的标记
- 每次暴力递归相当于子树内至少有一个标记消失
- 故一旦发生会使 T 减少至少 1

故总复杂度 $O(n \log n + q \log^2 n)$

原版证明可见 2016 国家集训队论文
或 https://codeforces.com/blog/entry/57319

Code Chef - Chef and Graph Queries

- 给定一棵树
- 若干此询问，给定 $[L, R]$，求仅保留编号在 $[L, R]$ 的边时连通块的数量

Sol

考虑离线, 从小到大枚举 R

注意到连通块数量 $= n-$ 最大生成森林边数

- 从小到大依次插入边
- 动态维护最大生成树边的编号集合

Sol

用 LCT 维护，对于一条即将加入的 (u, v) 编号为 R 的边

- 若 u, v 连通，断开路径上编号最小的边
- 连接 (u, v)
- Trick: 新建节点 R 连 $u+m$,$v+m$，维护点代替维护边

国家集训队矩阵乘法

给定一个 $n \times n$ 的矩阵，若干此询问
每次询问给定一个子矩形位置和 k ，求子矩形内第 k 大
$n \leq 500, q \leq 6 \times 10^4$

Sol

对矩阵元素离散化后，整体二分
记 $Calc(l,r,S)$ 为已知 $q \in S$ 询问的答案一定在 $[l,r]$ 内
用二维数组维护子矩形和

- 维护一个全局 k 表示恰好前 k 个位置被加入
- 每次只能 $k-1$ 或 $k+1$ 地移动
- 取 $mid = \lfloor \frac{l+r}{2} \rfloor$,
- 调整 $k = mid$ 对应位置插入数，查询矩阵和
- 所有询问递归被划分入 $[l, mid]$ 和 $[mid+1, r]$，递归处理

二维树状数组，单点修改 + 二维前缀和

```
void add(int x, int y, int d){
    for(int xx = x; xx <= n; xx += lowbit(xx))
        for(int yy = y; yy <= n; yy += lowbit(yy))
            a[xx][yy] += d;
}
void qry(int x, int y){
    int res = 0;
    for(int xx = x; xx > 0; xx -= lowbit(xx))
        for(int yy = y; yy > 0; yy -= lowbit(yy))
            res += a[xx][yy];
    return res;
}
```

$O(\log^2 n)$

Sol

- k 移动 $O(r-l)$ 次
  - 在进入 $[mid+1, r]$ 前将 k 移动至 mid
  - 总共 $O(n^2 \log n)$ 次
- 每个询问最多查询 $\log n$ 次
- 最终 $O((n^2+q)\log^3 n)$

ZJOI2019 线段树

- 给定一个区间赋值线段树的模板，维护一个操作序列
- 在序列尾部插入赋值操作
- 查询所有子序列操作完成后线段树中 tag 非空的节点数量和

```
function PushDown(Node)
    if tag[Node]= 1 then
        tag[Lson(Node)]← 1
        tag[Rson(Node)]← 1
        tag[Node]← 0
    end if
end function

function Modify(Node, l, r, ql, qr)
    if [l, r] ∩ [ql, qr] = ∅ then
        return
    end if
    if [l, r] ⊆ [ql, qr] then
        tag[Node] ← 1
        return
    end if
    m ← ⌊(l+r)/2⌋
    PushDown(Node)
    Modify(Lson(Node),l, m, ql, qr)
    Modify(Rson(Node),m + 1, r, ql, qr)
end function
```

Sol

考虑线段树每个节点维护 f_x 表示节点 x 有标记的方案数
维护 f_x 的和

Sol

问题转化为每个操作有 $\frac{1}{2}$ 的概率进行，求期望
考虑线段树每个节点维护 f_x 表示节点 x 有标记的概率
设 g_x 表示节点 x 到根路径上至少有一个节点有标记的概率
考虑一次 $[L, R]$ 赋值操作对区间 x 的影响

设 x 的父节点为 fa
若 x 父节点与 $[L, R]$ 无交，则对 x 无影响

$[L, R]$ 与 fa 有交，但与 x 无交
若此操作进行，则 x 有标记当且仅当 x 到根路径上至少有一个点有标记
$f_x \leftarrow \frac{1}{2}(f_x + g_x)$

Sol

$[L, R]$ 完全覆盖 x，未完全覆盖 fa 或 fa 不存在

- $f_x \leftarrow \frac{1}{2}(f_x + 1)$
- 对于 x 子树内的所有节点 y，$g_y \leftarrow \frac{1}{2}(g_y + 1)$
- 加上 tag 维护 g_x 值

Sol

$[L, R]$ 为完全覆盖 x

则此次操作后 x 到根路径上标记都会被清空

- $f_x \leftarrow \frac{1}{2} f_x$
- $g_x \leftarrow \frac{1}{2} g_x$

Sol

线段树上维护所有 f_x 的和，并用标记修改 g_x 最终答案为 $2^q \times \sum f_x$ 其中 q 为当前操作数

CF768G The Winds of Winter

- 给定一颗有根树，对于每个点 u，求以下值的最小值:
  - 将 u 和相关边删去，得到一个森林
  - 在剩余的有父亲的点中选择一个，称为 v
  - 将 v 和其父节点断开，然后将 v 连到任一节点上
  - 取所有连通块大小的最大值

Sol

考虑删掉一个点后，剩余的森林中树的大小
由于可以修改一次边，所以我们只关心

- 剩余子树大小的最大值 mx
- 剩余子树大小的次大值 mi
- 剩余子树大小的最小值 se

Sol

我们希望从从 mx 里挑出一个大小为 x 的子树，使得

- $\max\{mx - x, mi + x, se\}$ 最小
- 即令 x 最接近 $\frac{mx-mi}{2}$

Sol

- 维护一颗以子树大小为值域的线段树，线段树合并
- 如果 mx 是 u 的某个儿子, 直接求前驱后继即可

Sol

若 mx 在 u 父节点一侧，将其他点分为三类

- 是 x 的祖先
- 不在 x 的子树内也不是 x 的祖先
- x 的子树内

其中前两类是可能的 v,
三类点在 dfs 过程中各维护一个值域线段树
值域上做差分

- 给定一颗不平衡的线段树, 若干次询问
- 每次询问给定 u, l, r
- 找出构成线段树上构成 $[l, r]$ 的区间节点集合 S
- 求节点 u 到 S 中每一个节点的距离和

Sol

考虑如何找到所有的 S ? zkw 线段树!

Sol

- 从 $[l-1, l-1]$ 和 $[r+1, r+1]$ 向上找，直到到达 lca
- 从 $[l-1, l-1]$ 的节点的所有右兄弟
- 从 $[r+1, r+1]$ 的节点的所有左兄弟

Sol

把距离拆分成深度求和，差分
记录每个点到根所有节点的右/左兄弟的深度和
对 u 的具体位置情况进行讨论

- 在 lca 所在子树之外
- 在 lca 的左子树内
- 在 lca 的右子树内

洛谷 P6292 区间本质不同子串个数

- 给定一个字符串 S
- 每次询问 S 一个区间内本质不同的字符串数量

Sol

对询问按照 r 离线，建出后缀树

- 假设询问 l, r，考虑一个节点:
  - 字符串长为 $[L_l, L_2]$
  - 右端点集合为 e_i
- 取最大的 e_k 使得 $e_k \leq r$
- 贡献为 $[e_k - L_2, e_k - L_1]$ 与 $[l, r]$ 的交

Sol

从小到大枚举 r,

- 所有含 $e_i = r$ 的节点?
- $S[1, r]$ 对应节点在后缀树上到根的路径!
- 在后缀树上 *Access*
- 一条连续的路径先删去上一次 $e_k \neq r$ 的影响:
  - 每条 *splay* 上一次考虑的 e_k 相同
  - $[L_1, L_2]$ 的并仍然是一个区间
  - 每次 *splay* 在线段树上修改一次即可

给定一棵有根树，对每条边黑白染色，有若干个要求

- 每个要求给定 u, v，其中 u 是 v 的祖先
- 要求 u 到 v 路径上至少有一条黑边
- 求 2^{n-1} 个方案中有多少能满足所有要求
- 对 998244353 取模，$n = 5 \times 10^5$

Sol

容易想到进行树型 DP
对于 u，v 有很多方向 (不同子树)
但 v，u 只能在一个方向 (根)，故在 v 处所以要求
且注意到如果子树内有多个要求延申出子树，取最大深度 u 即可

Sol

设 $F(x,d)$ 为染完 x 子树，未满足 u 的最大深度为 d 的方案数

转移逻辑顺序：

- 合并儿子 y_1, y_2
  - $F(x, \max\{d_1, d_2\}) += F(y_1, d_1) \times F(y_2, d_2)$
- 处理完 x 的子树内，考虑是否染黑 x 到父节点的边

Sol

线段树动态开点维护 DP 值

- 合并儿子:
  - 线段树合并，过程中维护两棵树的前缀和
  - 遇到递归终点区间乘即可
- 考虑是否染黑到父节点的边:
  - 令 $F(x,0) += \sum_d F(x,d)$
  - 单点加

  注意处理完 x 后需要使得 $d \geq dep(x) - 1$ 的 $F(x,d) = 0$

给定一棵有根树，非叶子节点都有至少有 2 个节点

- 边有边权 $w_e > 0$，可以修改至任意非负数 w'_e
- 修改边权使得所有叶子到根距离相等
- 代价为 $\sum |w'_e - w_e|$

Sol

设 $f_x(d)$ 为仅考虑 x 子树和 x 到父节点的边，把所有叶子到 x 距离变为 d 的最小代价
设 $g_x(d)$ 为仅考虑 x 子树且不考虑 x 到父节点的边，把所有叶子到 x 距离变为 d 的最小代价

- 观察到 $f_x(d)$ 为下凸分段一次函数
- 斜率依次为 $-M, -M+1, -M+2, \cdots, 0, 1$
- 其中 M 为 x 子树内叶子数量
- 可能有某条直线退化为一个点

Sol

考虑极端情况

- 若想调整 $d = 1$
  - 令所有连叶子节点的边变成 1
- 若想调整称为 $+inf$
  - 花最小的代价使得 x 子树内叶子距离相同
  - 无限增加 x 到父节点边的边权

Sol

考虑仅维护拐点的可重集

- 把 x 的所有儿子的函数合并
  - $g_x \sum f_y$
  - 把拐点的可重集取并即可
  - 此时 g_x 在 $+\infty$ 处的斜率为 x 儿子的数量

Sol

考虑 g_x 变成 f_x 的影响

- 设 c 为 x 到父节点的边权
- 设 L, R 为 g_x 斜率为 0 直线两侧的拐点
    - 对于 $d > R + c$，一定有 f_x 斜率为 1
    - 对于 $d \in [L + c, R + c]$，一定有 f_x 斜率为 0
    - 对于 $d \in [L, L + c]$，一定有斜率为 -1
- 其他部分 f_x 与 g_x 斜率一致

故只需删除 $\geq L$ 的拐点，加入 $L + c$ 和 $R + c$ 两个拐点即可

Sol

数据结构维护一个可重集

- 合并
- 插入一个数
- 删除最大值

可并堆

如何求答案?

已知

- $f_{root}(0)$ 为所有点的边权和
- 相邻直线斜率相差 1

一些类似的题

- ZJOI2018 历史
- 清华集训 2015 V
- UOJ207 共价大爷游长沙
- ZJOI2019 语言

Thanks!