线段树的应用与推广

杨明天
浙江省镇海中学

2018年8月4日

目录

1 前言

在计算机科学中，数据结构是计算机中存储、组织数据的方式。因地制宜灵活运用各类数据结构可以提高算法的效率。

近年来，算法竞赛中高频出现数据结构有关内容。掌握并熟练运用各类数据结构，在当今时代显得尤为重要。

作为一类基础的数据结构，线段树在算法竞赛中有着十分重要的应用。同时，针对不同的问题，线段树也有着不同的推广。

本次讲课将以线段树为核心，向大家介绍线段树在维护序列问题、优化动态规划方面的应用，并推广到李超树、树链剖分等数据结构。以难度为序，辅以习题拓展。循序渐进，渐入佳境。希望能为各位同学在今后算法竞赛中的发展打下扎实的基础。

讲义中所选习题均能在网络上找到提交地址，对应的题解及源程序均能在我的博客[1] 中找到。

囿于本人才疏学浅，力有不逮，讲义中难免存在错误疏漏之处，恳请各位批评指正[2]。

2 学习目标

- 清楚了解线段树的原理
- 熟练掌握线段树的实现
- 能够合理利用线段树解决序列问题
- 能够使用线段树优化动态规划
- 能够使用李超树维护直线最值（凸壳）
- 能够使用线段树维护线段最值
- 能够使用树链剖分解决树上问题

3 用线段树解决序列问题

3.1 [JSOI2008]最大数

题目来源

[JSOI2008]最大数

题目大意

维护一个初始为空的数列，支持以下两种操作，操作共m次：

1. 查询当前数列中末尾k个数中的最大的数。
2. 将x插入到数列的末尾。

对于100%的数据，$m \le 2 \times 10^5$。

题目解法

直接维护整个长度为m的数列，单点修改，区间查询。

时间复杂度$\mathcal{O}(m \log m)$。

[1] https://skylee.gq/

[2] E-mail:skylee@skylee.gq

3.2 [TJOI/HEOI2016]排序

题目来源

[HEOI/TJOI2016]排序
[HDU5649]DZY Loves Sorting

题目大意

给定一个$1\sim n$的全排列，有m次操作，每次把区间$[l,r]$按照升序或降序排序。最后询问所有操作完成后，位置为q的数是多少。

对于100%的数据，$n,m\le 10^5$。

题目解法1

用桶排代替快排暴力模拟。不推荐采用这种做法。

题目解法2

题目只需要求位置为q的数是多少，而并不关心其他的数是多少。因此排序时也只需要考虑答案的那个数。

二分答案k，将$<k$的数当成0，$\ge k$的数当成1，原序列就变成了一个01序列。

排序时只需要统计区间内0 和1的个数，线段树区间修改即可。

若排序后q上的值为1，则答案$\ge k$，否则$<k$。

时间复杂度$\mathcal{O}(m\log^2 n)$。

题目解法3

另外还有一种$\mathcal{O}(m\log n)$的在线做法。

用可分裂合并的线段树,即建n个值域线段树，然后在线做。

用set维护每次要合并的区间，如果区间$[l,r]$“切断”了一个已经合并的区间，就把它分裂。

具体做法参考这篇题解。

3.3 [POJ2104]K-th Number

题目来源

[POJ2104]K-th Number

题目大意

给一串n个数a_i，m次询问区间k小值。

对于100%的数据，$n\le 10^5,m\le 5000,|a_i|\le 10^9$。

题目解法1

归并树：用线段树维护数列在归并排序每一阶段状态的数据结构。每个结点存储一个数组。

首先构建一棵归并树，然后二分查找最小的数x 使得被查找区间中$\le x$的数的个数大于等于$\ge k$。

时间复杂度$\mathcal{O}(m\log^3 n)$。

题目解法2

建立一棵主席树，查询时在主席树上二分即可。
时间复杂度$\mathcal{O}(m\log n)$。

题目解法3

整体二分+树状数组。
时间复杂度$\mathcal{O}(n\log n\log \mathrm{range}(a_i))$。
具体做法参考这篇题解。

3.4 [NOI2017]整数

题目来源

[NOI2017]整数

题目大意

n次操作维护一个长度为$30n$的二进制整数x，初始为0，支持以下两种操作：

1. 将这个整数加上$a_i\cdot 2^{b_i}$。
2. 询问这个整数第k_i位的值。

对于100%的数据，$n\le 10^6,|a_i|\le 10^9,b\le 30$，保证任何时刻$x\ge 0$。

题目解法

维护每一位的值，并在线段树上记录每个区间是否含有0或1，以便发生进退位时快速查找到进退位结束的位置。区间修改，单点查询。由于a 最多有$\log a$个二进制位，因此时间复杂度$\mathcal{O}(n\log(30n)\log a)$。显然会TLE。

考虑使用压位线段树，每30位压在一个叶子结点，修改操作可以枚举a的每一个二进制位1 并进行相应的操作。时间复杂度$\mathcal{O}(n\log n\log a)$。实测76分。

事实上，由于a在线段树中最多对应2个叶子结点，而叶子结点内部的进位就是整数加减法，不需要专门在线段树上查找。我们可以直接将a 拆分成两次修改，时间复杂度$\mathcal{O}(n\log n)$。

3.5 [NOI2016]区间

题目来源

[NOI2016]区间

题目大意

有n次操作，每次覆盖数轴上的区间$[l_i,r_i]$。

现在要你挑出m次操作，使得数轴上有一个整点恰好被覆盖m次，且最大覆盖区间与最小覆盖区间大小之差最小。

对于100%的数据，$n\le 5\times 10^5,m\le 2\times 10^5,0\le l_i,r_i\le 10^9$。

题目解法

把询问按长度排序，用尺取法$\mathcal{O}(n)$枚举左右端点，用线段树维护每个点被覆盖了几次。

时间复杂度$\mathcal{O}(n\log n)$。

3.6 [NOIp2017提高组]列队

题目来源

[NOIp2017提高组]列队

题目大意

一个$n\times m$的方阵，每个格子里的人都有一个编号。初始时第i行第j列的编号为$(i-1)\times m+j$。

q次事件，每次在(x,y)位置上的人离队。剩下的人向左、向前填补空位，然后离队的人在(n,m)处归队。

求每次离队事件中的人的编号。

对于100%的数据，$n,m,q\le 3\times 10^5$。

题目解法

对于每一行的$1\sim m-1$列建一棵线段树，对于最后一列也建一棵线段树。开同样数量的vector。

(x,y)离队时，在第x棵线段树上找到第y个未移动的值在vector中的位置，再从最后一列的线段树中找到第x个未移动的值加入第x个vector末尾，最后将答案加入最后一列对应vector末尾即可。

时间复杂度$\mathcal{O}(q\log n)$。

3.7 [POI2014]Karty

题目来源

[POI2014]Karty

题目大意

有n张卡片排成一排，每张卡片正反面有两个数a_i和b_i。m次操作，每次交换第c_i和第d_i张卡片，问若可以任意翻转卡片，是否存在一种方案使得卡片上的数字构成一个不下降序列。

对于100%的数据，$n\le 2\times 10^5, m\le 10^6$。

题目解法

显然，对于一个区间，当左端点是一个固定的值时，右端点尽量取小的。

用线段树维护区间，左端点取较大/小值时，右端点取较大值还是较小值，还是无论右端点取较大值或较小值，都无法保证单调不降。

对于每次交换的操作，在线段树上修改即可。

3.8 [CF580E]Kefa and Watch

题目来源

[CF580E]Kefa and Watch

题目大意

一个由'0'～'9'构成的长度为n的字符串，支持以下两种操作，操作共m次：

1. 将区间内$[l, r]$的所有字符修改为d。
2. 询问区间$[l, r]$是否有长度为d的循环节。

对于100%的数据，$n, m \le 10^5$。

题目解法

线段树维护字符串哈希值。

若$r - l + 1 \le d$，则区间$[l, r]$一定有长度为d的循环节，否则比较区间$[l, r-d]$和区间$[l+d, r]$ 的哈希值即可。

3.9 [清华集训2014]奇数国

题目来源

[清华集训2014]奇数国

题目大意

一个长度为10^6的数列，支持以下两种操作，操作共m次：

1. 修改某一点的数。
2. 求某一区间乘积的欧拉函数。

保证每个元素的最大质因数不超过281，答案对19961933取模。

对于100%的数据，$m \le 10^5$。

题目解法

$\varphi(n) = n(1-\frac{1}{p_1})(1-\frac{1}{p_2})...(1-\frac{1}{p_l})$，其中$p$为$n$的不同的质因数。

可以发现欧拉函数值只与n和p有关。

所以可以用线段树维护每个区间的乘积，因为281是第60个质数，因此可以二进制存储每个质数是否出现。

最后套用欧拉函数公式进行计算即可。

3.10 [BZOJ4964]加长的咒语

题目来源

[BZOJ4964]加长的咒语

题目大意

给你一个长度为n的括号序列s，不一定合法。

有m组询问，每次询问一段区间$[l, r]$中最长的合法括号序列的长度。

对于100%的数据，$n, m \le 4 \times 10^5$。

题目解法

首先我们可以将'('看做-1，将')'看做$+1$，那么我们可以求出这个括号序列的前缀和。

不难发现对于其中一个合法的子串$(l, r]$，其左右括号的数目相等，因此两端所对应的前缀和是一样的（然而反之则不一定）。

对于每一组询问，我们可以找出前缀和最大值max，显然一个合法的括号序列要么与max无关，要么刚好两端都是max。

我们可以求出max在区间内出现的最左和最右的位置p_1和p_2，那么答案要么就是$(p_1, p_2]$，要么就是$[l, p_1]$或$(p_2, r]$中最长的合法序列。

这时候我们就可以对于两个区间分别求一下往左、往右能扩展的最大距离，将其作为答案即可。这里可以用线段树或者稀疏表实现。

时间复杂度$\mathcal{O}(m \log n)$。

3.11 [POI2015]Kinoman

题目来源

[POI2015]Kinoman

题目大意

给你一个长度为n的数列f，f中共有m种不同的数，每种数都有一个权值$w[i]$。

你可以选定一个f中的区间，定义区间的权值为这一区间只出现一次的数的权值和。

问权值最大的区间的权值是多少？

对于100%的数据，$n, m \leq 4 \times 10^6$。

题目解法

对于f中的每一个位置i，找到下一个和它相同数字的位置$next[i]$。

从左到右枚举区间左端点，线段树维护选取每个右端点的最大值。

去除当前左端点对答案的影响时，只需要把$i \sim next[i]-1$这一段减去$w[f[i]]$，然后把$next[i] \sim next[next[i]]-1$加上$w[f[i]]$即可。

3.12 [九省联考2018]IIIDX

题目来源

[九省联考2018]IIIDX

题目大意

一个游戏有n个关卡，每个关卡有一个难度系数d_i。给定一个实数$k(k \leq 10^9)$，表示关卡i依赖于关卡$\lfloor \frac{i}{k} \rfloor$，即只有当关卡通过$\lfloor \frac{i}{k} \rfloor$后，关卡$i$才会被解锁。问如何排列这些难度系数，使得$d_i \geq d_{\lfloor \frac{i}{k} \rfloor}$？求字典序最大的方案。

对于100%的数据，$n \leq 5 \times 10^5, k, d_i \leq 10^9$。

题目解法

不难发现关卡间的依赖关系构成了一个树状结构。

对于d_i各不相同的情况，只存在唯一的一种方案，直接贪心即可，期望得分60分。

对于d_i相同的情况，用权值线段树维护大于当前难度值有多少未分配的难度值，从小到大枚举$1 \sim n$号点，对于第i号结点，将当前未分配的第$size[i]$大的难度值分配给该结点即可。

考虑如何操作才能使得d_i之后的难度值一定分配给i的子树。

在操作完i结点后，可以将前$size[i]$大的难度值在线段树上标记为已分配，当访问到i的第一个子结点时，再恢复这些难度值并进行分配。

3.13 [TC14133]FindingFriends

题目来源

[TC14133]FindingFriends

题目大意

给定一个长度为n的数列A，求最小的k满足存在一个长度至少为m的子串，对于串中的每一个数A_i，都至少存在一个$A_j(i \neq j)$满足$|A_i - A_j| < k$。

对于100%的数据，$n \leq 10^5, A_i \leq 10^9$。

题目解法

二分答案k，对于每次求出每个元素A_i左边离A_i最近的满足$|A_i - A_j| < k$的$left[i] = j$，同理求出每个元素A_j右边离A_i最近的满足$|A_i - A_j| < k$的$right[i] = j$。

考虑分治，判断区间$[l, r]$是否是满足条件的子串。

若$r - l + 1 < m$则显然不满足条件。

在$[l, r]$中找到一个下标p满足$left[p] < l$且$right[p] > r$，则若这样的p存在，则包含p的区间一定不满足条件，递归处理$[l, p)$和$(p, r]$。若找不到这样的p，则该区间满足条件，k为合法的答案。

寻找p时，如果不是直接for而是从两端同时往里寻找，设找到的p两端的区间长度分别是a和b，则递归的复杂度是$\mathcal{O}(n \log n)$（递推式$T(n) = T(a) + T(b) + \min(a, b)$）。而$left$和$right$的预处理可以用线段树实现，总的时间复杂度是$\mathcal{O}(n \log^2 n)$。

3.14 [CF903G]Yet Another Maxflow Problem

题目来源

[CF903G]Yet Another Maxflow Problem

题目大意

有A类点和B类点各n个，所有A_i到A_{i+1}有一条权值为a_i的有向边，所有B_i到B_{i+1}有一条权值为b_i的有向边，另有m条从A_x到B_y的权值为有向边。连续q次操作将A_{v_i}与A_{v_i+1}之间的边的权值改为w_i。问每次修改完毕后的从A_1到B_n的最大流。

对于100%的数据，$n, m, q \leq 2 \times 10^5$。

题目解法

根据最大流-最小割定理，题目所求相当于每次修改完毕后的最小割。定义A类点间的边为A类边，B类点间的边为B类边，AB类点间的边为C类边。假设两类边各$n-1$条之外还分别有一个边权为0的边，那么每次的最小割一定恰好包含一个A类边、一个B类边和若干C类边。由于B类边和C类边都不会被修改，则对于同一个A类边，对应的最优的B类边是固定的。方便起见，下文将B_i到B_{i+1}的边记作b_{i+1}，不同于题面描述。

考虑预处理每个A类边对应的最优B类边。不难发现，若我们选择了a_i和b_j两条边，要使得A_1与B_n不连通，则我们还需要割去所有连接$A_x, B_y(x \le i, y \ge j)$的$C$类边，而这也是选择$a_i$和$b_j$后的最小割。反过来说，连接$A_x, B_y$的$C$类边会对$a_i, b_j(x \le i, y \ge j)$ 的选择产生影响。因此我们可以$1 \sim n$枚举每个a_i，用线段树维护对应每个b_j所需要的最小割的大小。首先将所有b_j加入到线段树中，对于当前枚举到的a_i，枚举从A_i出发的所有C类边，若对应的点为B_j，权值为w，将区间$[1, j]$加上w，表示对于a_i及a_i以后的A类边，若还要考虑b_j及b_j以前的B类边作为对应边，一定要割去这条C类边。而每次插入后线段树最小元素就是对应当前a_i，由一个B类边和若干C类边组成的、能与a_i构成割的边权和，记这一边权和为sum，则选择a_i时的最小割为$sum + a_i$，记作c_i。

考虑修改操作，由于a_i对应的B类边和C类边已经确定，每次修改时a_i的变化也就是c_i的变化。用一些数据结构维护所有c_i的最小值即可。时间复杂度$\mathcal{O}((m+q)\log n)$。

3.15 [HNOI/AHOI2018]转盘

题目来源

[HNOI/AHOI2018]转盘

题目大意

一个环上有n个物品。在时间为0的时候，你可以任选一个点作为起点出发。每秒钟你可以选择留在当前点或走到下一个点。每个物品有一个出现的时间t_i。对于每一时刻，若当前位置上的物品已经出现了，则可以获得该物品。问何时可以获得所有物品。

另有m次修改操作，对于每次修改$t_x = y$，求出修改后的答案。强制在线。

对于100%的数据，$n, m \le 10^5$。

题目解法

将环复制一遍接在n的后面，变成一个链。

设起点$i \in [n, 2n)$，往前走到$j \in (i-n, i]$。设我们在时刻s到达j，并获得j上的物品。那么我们可以知道$s - (i-j) \ge t_j$，因此$s_{\min} = \max\{t_j - j\} + i$。

令$a_i = t_i - i$，$ans = \min\limits_{i \in [n, 2n)} \{\max\limits_{j \in (i-n, i]} a_j + i\} = \min\limits_{i \in [1, n]} \{\max\limits_{j \in [i, 2n]} a_j + i\} + n - 1$。使用线段树维护单调栈即可。

时间复杂度$\mathcal{O}(n \log^2 n)$。

3.16 [HNOI/AHOI2017]影魔

题目来源

[HNOI/AHOI2017]影魔

题目大意

有一排n个数$k_{1\sim n}$。对于点对(i,j)，若不存在$k_s(i<s<j)$大于k_i或k_j，则对答案造成p_1的贡献；若$c=\max_{s\in(i,j)}\{k_s\}$满足$k_i<c<k_j$或$k_j<c<k_i$则对答案造成$p_2$的贡献。$m$次询问，每次询问区间$[l,r]$内所有点对对答案贡献之和。其中$k_i$为$1\sim n$的全排列。

对于100%的数据，$n,m\le 2\times 10^5$。

题目解法

首先预处理出对于每个点i，其左侧第一个权值大于它的点$left[i]$和其右侧第一个权值大于它的点$right[i]$。显然这个点i对答案的贡献有3种情况：

1. 对于点对$(left[i],right[i])$，贡献为p_1；
2. 对于所有点对$(l\in(left[i],i),right[i])$，贡献为$p_2$；
3. 对于所有点对$(left[i],r\in(i,right[i]))$，贡献为$p_2$。

我们可以离线处理所有询问。将询问和贡献分别排序，用树状数组维护答案即可。

3.17 [BZOJ4373]算术天才⑨与等差数列

题目来源

[BZOJ4373]算术天才⑨与等差数列

题目大意

一个长度为n的数列$\{a_i\}$。m次操作，操作包含以下两种：

1. 问区间$[l,r]$内的数从小到大排序后能否形成公差为k的等差数列。
2. 修改数列的某一项。

强制在线。

对于100%的数据，$n,m\le 10^3$。

题目解法

考虑形成等差数列的条件：

1. 区间内所有数差分的$\gcd=x$。
2. 区间$\max-\min=(r-l)\times k$。
3. 区间内数字不相同。

线段树维护最大、最小值以及差分。

对于每次询问判断上述三个条件，如果满足则说明可以构成等差数列。

4 用线段树优化动态规划

4.1 [CF115E]Linear Kingdom Races

题目来源

[CF115E]Linear Kingdom Races

题目大意

有n个物品，编号为$1 \sim n$。选取第i个物品需要c_i的代价。另外有m个条件，表示若$l_i \sim r_i$间的物品全部选择，可以获得p_i的收益。求最大收益。

对于100%的数据，$n, m \le 10^5$。

题目解法

用$f[i]$表示考虑完前i个物品是否选取能获得的最大收益。

转移方程为$f[i] = \max\{f[j] - \texttt{cost}(j+1, i) + \texttt{profit}(j+1, i)\}$。

使用线段树优化即可。

时间复杂度$\mathcal{O}((n+m)\log n)$。

4.2 [ARC073F]Many Moves

题目来源

[ARC073F]Many Moves

题目大意

有一排n个格子和2枚硬币。

有q次任务，每一次要你把其中一枚硬币移到x_i的位置上，移动1格的代价是1。

两枚硬币不能同时移动，任务必须按次序完成。

告诉你两枚硬币初始状态所在的位置a和b，问完成所有任务的最小代价。

对于100%的数据，$n, q \le 2 \times 10^5$。

题目解法

由于完成任务的次序确定，每个任务的位置也确定，我们可以用$f[i][j]$表示完成第i个任务后，一个硬币在x_i，一个硬币在j的最小代价。

转移方程为:

$$\begin{cases} f[i][j] = \min\{f[i-1][j] + |x_i - x_{i-1}|\} \\ f[i][x_{i-1}] = \min\{f[i-1][j] + |x_i - j|\} \end{cases}$$

时间复杂度$\mathcal{O}(qn)$。

不难发现，在状态转移方程中，如果我们能去掉绝对值，里面的东西就能用线段树维护。

而绝对值的取值只与硬币的左右位置关系有关。

因此我们可以建两棵线段树，一棵表示被转移的状态在目标状态左边，一棵表示在右边。

左线段树中每个叶子结点x_{i-1}维护$f[i-1][j] - x_{i-1}$的值，右线段树每个叶子结点x_{i-1}维护$f[i-1][j] + x_{i-1}$的值。

时间复杂度$\mathcal{O}(q \log n)$。

5 用李超树维护直线最值（凸壳）

5.1 [CSA]Squared Ends

题目来源

[CSA]Squared Ends

题目大意

给你一个长度为n的数列$\{A_i\}$。定义区间$A_{[l,r]}$的代价为$(A_l - A_r)^2$。求将$\{A_i\}$划分成k个区间的最小代价。

对于100%的数据，$n \le 10^4, k \le 100, A_i \le 10^6$。

题目解法

不难想到一种动态规划，用$f[i][j]$表示已经划分了i个区间，结尾是j的最小代价。转移方程为:$f[i][j] = \min\{f[i-1][k-1] + (A_j - A_k)^2\}$

时间复杂度是$\mathcal{O}(n^2k)$，显然会TLE。

变形得：$f[i][j] = A_j^2 + \min\{-2A_jA_k + f[i-1][k-1] + A_k^2\}$

其中min中的东西可以看做是关于A_j的一次函数。而寻找min值的过程就相当于在一堆一次函数中找最小值，用李超树维护凸壳即可。

时间复杂度$\mathcal{O}(nk \log \mathrm{range(A_i)})$。

5.2 [ARC051D]長方形

题目来源

[ARC051D]長方形

题目大意

给定$A_{1\sim n}$和$B_{1\sim m}$，矩阵$C_{i,j} = A_i + B_j$。q次询问，求坐标不超过(x, y)的最大权值子矩阵的权值。

对于100%的数据，$n, m, q \le 2000, |A_i|, |B_i| \le 10^5$。

题目解法

用$maxa[i][j]$表示$A_{1\sim i}$中长度为j的最大连续子段和，$maxb$同理。

对于询问(x, y)，答案即为$\max\limits_{i \le x, j \le y}\{maxa[x][i] \times j + maxb[y][j] \times i\}$。

max内变形后即为$j \times (maxa[x][i] + \frac{maxb[y][j] \times i}{j})$。括号内的东西可以看作是关于$\frac{maxb[y][j]}{j}$的一次函数，使用李超树维护凸壳即可。

时间复杂度$\mathcal{O}(qn \log n)$。

5.3 [YC703]ゴミ拾い Easy

题目来源

[YC703]ゴミ拾い Easy

题目大意

二维平面内有n个人和n个物品，第i个人在$(a_i,0)$上，第i个物品在(x_i,y_i)上，满足$a_i<a_{i+1},x_i<x_{i+1}$。每个人可以取走一些物品或者一个也不取。第i个人取走物品$j\sim i$的代价为这个人到物品j的欧几里得距离的平方。每个人不能取比自己编号大的物品，问所有物品都被取完的最小代价。

对于100%的数据，$n\le 3\times 10^5, 0\le a_i,x_i,y_i\le 10^5$。

题目解法

用$f[i]$表示前i个人取走前i个物品的最小代价，一个显然的DP为：$f[i]=\min\limits_{0\le j<i}\{f[j]+(x_{j+1}-a_i)^2+y_{j+1}^2\}$。

将min中间的展开，就是$a_i^2+(-2x_{j+1}a_i+x_{j+1}^2+y_{j+1}^2+f[j])$。括号内可以看作是一个关于$a_i$的一次函数。使用李超树维护一次函数最小值即可。

时间复杂度$\mathcal{O}(n\log n)$。

5.4 [BZOJ4700]适者

题目来源

[BZOJ4700]适者

题目大意

有n个敌人，每个敌人有一个攻击力v_i和一个防御力d_i。你的攻击力是a。战斗看做回合制，每回合进程如下：

1. 你选择某个敌人进行攻击，令其防御力减少a，若防御力<0则该敌人被击败。
2. 所有存活的敌人每人对你造成v_i点损失。

战斗开始前你有机会直接去掉对方的两个敌人。

你拥有无限的血量，请最小化你的损失。

对于100%的数据，$n\le 3\times 10^5, v_i,d_i,a\le 10^4$。

题目解法

首先不考虑去掉两个敌人的情况。

显然我们可以按一定顺序依次击败各个敌人，而不必一个人打一半就去打其他人。

用t_i表示击败敌人i所花的时间，则$t_i=\lceil\frac{d_i}{a}\rceil$。

考虑攻击的顺序。i在j前面，当且仅当$t_i\times v_j<t_j\times v_i$。

此时$ans=\sum_{i=1}^{n}(\sum_{j=1}^{i}t_j-1)\times v_i$。

现在考虑事先去掉两个敌人的情况，即如何选择去掉的敌人使损失最小。

用pre表示t_i的前缀和，用suf表示v_i的后缀和。

考虑只去掉一个敌人的情况，去掉i这个敌人可以使答案减少$c_i=suf[i]\times t_i+(pre[i-1]-1)\times v_i$。

而去掉两个人就是要找到一对i,j，使得$c_i+c_j-v_i\times t_j$最大。

考虑固定一个i，则j的贡献就是一个关于v_i的一次函数，用李超树维护凸壳即可。

时间复杂度$\mathcal{O}(n\log n)$。

6 用李超树维护线段最值

6.1 [CC-STREETTA]The Street

题目来源

[CC-STREETTA]The Street

题目大意

给定两个长度为n的数列A和B，开始数列A中每一项值为$-\infty$，数列B中每一项值为0。m次操作，操作包含以下3种：

1. 数列A区间加一条等差数列。
2. 数列B区间对一个等差数列取max。
3. 询问$A_i + B_i$。

对于100%的数据，$n \leq 10^9, m \leq \times 10^5$。

题目解法

每个结点维护一个解析式$kx + b$。

对于数列A，使用李超树维护最大值。

对于数列B，直接合并两个解析式。

时间复杂度$\mathcal{O}(m \log n)$。

7 用树链剖分解决树上问题

7.1 [JLOI2014]松鼠的新家

题目来源

[JLOI2014]松鼠的新家

题目大意

一棵n个结点的树，按照给定的顺序访问结点。

每次访问将路径上经过结点的点权+1,问最后各个结点的点权。

对于100%的数据，$n \leq 3 \times 10^5$。

题目解法

树链剖分后用线段树维护每个点的权值。

时间复杂度$\mathcal{O}(n \log^2 n)$。

7.2 [NOI2015]软件包管理器

题目来源

[NOI2015]软件包管理器

题目大意

给你一棵n个结点的树，q次操作，操作包含以下两种：

1. 把根到x路径上的所有点的点权都变成1。
2. 把x的子树中，所有点的的点权都变成0。

问每一步操作时，有多少点的点权发生了改变。

对于100%的数据，$n, q \le 10^5$。

题目解法

子树在树链剖分序中对应的是一个完整的区间。

到根路径在树链剖分序中对应的是$\mathcal{O}(\log n)$个区间。

线段树区间修改区间查询，时间复杂度$\mathcal{O}(n \log^2 n)$。

7.3 [CEOI2017]One-Way Streets

题目来源

[CEOI2017]One-Way Streets

题目大意

给你一个n个点，m条边的无向图，告诉你p个点对(u, v)，要你在保证从u到v的所有路径都不变的情况下，尽可能将所有的边确定方向。

问你可以唯一确定哪些边的方向，以及方向是从u到v还是从v到u。

对于100%的数据，$n, m, p \le 10^5$。

题目解法

不难发现环上的边都不能确定方向，所以我们可以先缩环。

缩环以后剩下的图就变成了一棵树。

树链剖分，用线段树维护边的方向。

时间复杂度$\mathcal{O}(p \log^2 n)$。

7.4 [LOJ6208]树上询问

题目来源

[LOJ6208]树上询问

题目大意

有一棵n节点的树，根为1号节点。每个节点有两个权值k_i, t_i，初始值均为0。

m次操作，操作包含以下三种：

1. Add(x, d)操作：将x到根的路径上所有点的$k_i \leftarrow k_i + d$。
2. Mul(x, d)操作：将x到根的路径上所有点的$t_i \leftarrow t_i + d \times k_i$。
3. Query(x)操作：询问点x的权值t_x。

对于100%的数据，$n, m \le 10^5$。

题目解法

树链剖分以后用线段树维护。

对于每个结点，我们可以维护3个数a, b, c，表示最后$t = a \times b + c$。

对于操作1，需要修改a和c。

对于操作2，需要修改b。

然后操作3就变成了单点查询。

时间复杂度$\mathcal{O}(m \log^2 n)$。

7.5 [HNOI2016]网络

题目来源

[HNOI2016]网络

题目大意

给你一棵n的数，有m次操作，操作包含以下三种：

1. 新建一个从u到v的任务，权值为w。
2. 删除第i个任务。
3. 询问所有不经过x的任务中最大的权值。

对于100%的数据，$n \le 10^5, m \le 2 \times 10^5$。

题目解法

首先对树进行轻重链剖分，建立线段树。

每个线段树结点套二叉堆，维护不经过这个结点的权值。

对于操作1，我们可以先求LCA，同时求出路径上经过的所有链，修改没有经过的那一些链。

对于操作2，考虑另建一个堆来维护删除的那些权值。

对于操作3，在取top 时，先比较一下两个堆的top是否相同，如果相同就忽略掉。

时间复杂度$\mathcal{O}(m \log^2 n \log m)$。

7.6 [CF160D]Edges in MST

题目来源

[CF160D]Edges in MST

题目大意

一个n个点，m条边的连通图。对于图中的每条边，判断它与该图最小生成树的关系：

1. 在该图所有的最小生成树中；
2. 在该图至少一个最小生成树中；
3. 不在该图的任何一个最小生成树中。

对于100%的数据，$n, m \le 10^5$。

题目解法

首先用Kruskal求出该图其中一个最小生成树。

枚举每一条不在树内的边，考虑在生成树上加入这条边所构成的环。

若环上有权值与该边相同的边，则这些边都是可以互相替换的，都属于第2类关系。

若环上所有边的权值都比该边小，则说明包含该边的生成树不是最小的，属于第3类关系。

判断完所有的2、3类关系后，生成树中剩下的边即为1类边。

显然加入非树边时，环上不存在权值大于该非树边的边，否则与最小生成树矛盾。因此可以考虑使用树链剖分和线段树维护最大值。

时间复杂度$\mathcal{O}((n+m)\log n)$。

7.7 [CC-QUERY]Observing the Tree

题目来源

[CC-QUERY]Observing the Tree

题目大意

给定一棵n个节点的树，m次操作，操作包含以下三种：

1. 路径加等差数列。
2. 询问路径和。
3. 恢复到第i次修改后的情况。

强制在线。

对于100%的数据，$n,m\le 10^5$。

题目解法

首先不考虑操作3，对树进行树链剖分，用线段树维护。

对线段树上每一个节点用a,b两个标记，表示对这个区间加上首项为a，公差为b的等差数列，这是可以合并的。

因为有第三种操作，所以要使用可持久化线段树。

时间复杂度$\mathcal{O}(m\log^2 n)$。

7.8 [CC-QTREE]Queries on tree again!

题目来源

[CC-QTREE]Queries on tree again!

题目大意

给定一棵n个节点的带边权的基环树，两点之间的最短路径定义为经过点数最少的路径。

保证环的大小是奇数，即保证任意两点的最短路径唯一。

m次操作，操作包含以下两种：

1. 把两点之间最短路径上的所有边的权值取相反数。
2. 询问两点之间最短路径上的边权的最大子段和。

对于100%的数据，$n,m\le 10^5$。

题目解法

对每一棵外向树维护一个树链剖分，然后再对环上所有边维护一棵线段树。

于是每一次修改和询问都可以转化成$\mathcal{O}(\log n)$次线段树上的操作。

在线段树的每个节点上维护最大/最小子段和，前缀/后缀最大/最小和就好了。

时间复杂度$\mathcal{O}(m\log^2 n)$。

7.9 [CC-QTREE6]Query on a tree VI

题目来源

[CC-QTREE6]Query on a tree VI

题目大意

给定一棵n个节点的树，每一个节点都是黑色或者白色。

m次操作，操作包含以下两种：

1. 修改一个点的颜色。
2. 如果只保留树上连接两个相同颜色点的边，询问一个点所在联通块的大小。

对于100%的数据，$n, m \leq 10^5$。

题目解法

我们可以对黑色和白色两种颜色分别维护一个树链剖分，对于每一个节点维护假定它是黑色/白色且只考虑它的子树时它所在黑色/白色联通块大小。

这可以直接用传统的维护子树和的方法维护，修改颜色的时候只需要在两条树链剖分上更新修改节点到根的路径就行了，询问的时候就找到距离它最远的，路径上都是和它同色点的祖先，然后查询树链剖分上的单点值即可。

要支持的操作只有区间加，用树状数组维护即可。

时间复杂度$\mathcal{O}(m\log^2 n)$。