数据结构

黄洛天

THU，IIIS

April 27, 2025

一些要讲的：

- 平衡树相关应用
- top tree

top tree 这种东西真的有必要学吗.jpg
在此之前，我们先讲点 useful 的东西。

gym 104090 B

https://codeforces.com/gym/104090/problem/B

gym 104090 B

gym 104090 B

因为式子里全是 max，所以你可以考虑每一位的贡献然后取 max，因此考虑对于每一位求出最大的 $d_i + d_j$ 满足第 i 位会进位。

令 $A = c_i \bmod 2^i, B = c_j \bmod 2^i$ 考虑枚举每一位，判断 $a + b$ 是否会在第 i 位进位只需要判断 $A + B$ 是否大于等于 2^i 即可。如果我们把 A, B 当下标 f_A 表示值为 A 的元素里 d 最大的。

那么就相当于查询 $\max_{i+j \le 2^i} f_i + f_j$，也就是一个 $(max, +)$ 卷积的形式。

这个形式不好看，令 $g_A = F_{2^i - A}$，那么相当于查询 $\max_{i \le j} g_i + f_j$。这个在线段树上可以方便的维护，每个节点维护区间内 $\max_{i \le j} g_i + f_j, \max_i g_i, \max_i f_i$，即可。

时间复杂度 $O(n \log n \log w)$。

gym 103861 F

https://codeforces.com/gym/103861/problem/F

gym 103861 F

对序列按照 C 分段，分别考虑两类跨段的贡献和不跨段的贡献。

考虑每个块维护一个线段树，用来处理单点修改，区间最大子段和。那么不跨块的贡献由 l, r 所在块的 $1, 2$ 两次区间查询，以及在 l, r 直接的块的全局 max，所以额外一个线段树单点修改区间 max，维护所有块整块的最大子段和。

对于跨块的情况，令 f_i 表示前一个块的倒数 i 个数字的后缀和，g_j 表示后一个块的前 j 个数字的和。那么就是要求 $\max_{i+j\le C} f_i + g_j$，可以用上一题完全一样的方法解决。同样需要特殊查询 l, r 所在块的贡献，以及中间完整的块的贡献。

时间复杂度 $O(n \log n)$。

uoj 515

https://uoj.ac/problem/515

uoj 515

考虑离线扫描线序列，维护每个询问。

对于每个询问有一个目前的最小值 min，以及目前的答案 ans。

对于某个序列的上的位置，这个位置被单点修改划分为了若干个时间段，每个时间段的值是不一样的，设某个时间的值为 x，对于这个时间的询问需要做的事情是：
$if(x < min) min = x, ans++;$。

事实上这个正式 beats 的形式，考虑维护区间最大值和严格次大值。如果 $secmax < x < max$，那么需要给所有最大值的位置 $ans++$，并且让 max 变成 x。

pushdown 的时候注意 ans 是只给区间最大值才下发的标记。

时间复杂度 $O(n \log n)$。

gym 104172 I

https://codeforces.com/gym/104172/problem/I

首先介绍一个线性的平面最近点对做法。考虑增量法，每次加入第 i 个点，假设前 $i-1$ 个点的答案为 ans，把平面按照 $([\frac{x}{ans}],[\frac{y}{ans}])$ 分块。如果新加入的点能比之前的 ans 更优秀，那么另一个点一定在新加入的点所在块的 3×3 相邻块里，因为每个块内部点数为常数，所以只会访问 $O(1)$ 个点。如果提前对点 shuffle，那么加入第 i 个点时成为新最优解的概率是 $\frac{1}{i}$，如果成为了最优解重构分块的的代价是 $O(i)$，总时间复杂度 $O(n)$。

gym104172l

考虑扩展上述做法，我们强制只能对平面做 2^i 分块。可以从右到左扫描线每次加入一个点 i，对于相邻块里每个点 j 计算一下距离，并在数据结构的 j 处对 $dis(P_i, P_j)$ 取 min，每次查询就是查询一个前缀 min。但是上面的做法因为没有重构导致每个块内点数并不是 $O(1)$ 的。

令 $f_i(l)$ 表示对平面 2^i 分块，区间 $[l, r](r > f_i(l))$，能找到一对点使得他们在相邻块里。第 2^i 分块里能找到的最远点对是 $2^{i+1}\sqrt{2}$。能新找到的最近点对距离是 2^i。因为有 $2^{i+1}\sqrt{2} < 2^{i+2}$，因此第 2^{i+2} 分块枚举到的点对不会更新 $[l, f_i(l)]$ 或更长的区间的答案。也就是说我们扫描线的时候可以做一个双指针，扫到 l 的时候只保留 $[l, f_{i-2}(l)]$ 里的点，这样的话每个块内点数就是 $O(1)$ 了。时间复杂度 $O(n \log n \log w)$。

loj3273

在 $x+y\leq n, x, y\geq$ 的平面上初始有 m 个点，维护一个数据机构支持 4 种操作：

1. 查询第 k 次插入的点现在的位置。
2. 给一个 l，有个初始在 y 轴长度为 l 的线段向右移动推点走到 $n-l$，形式化的说 $(x,y), x\leq n-l, y\leq l$ 变为 $(n-l,y)$。
3. 给一个 l，有个初始在 x 轴长度为 l 的线段向上移动推点走到 $n-l$，形式化的说 $(x,y), x\leq l, y\leq n-l$ 变为 $(x,n-l)$。
4. 插入一个点 (x,y)。

$n\leq 10^9, m\leq 5\times 10^5, q\leq 10^6, TL=11s, ML=1G$

loj3273 - Solution

注意到一个点一旦被碰到就会永远落在操作的折线上，并且每次操作不会影响折线上点的相对位置，对折线的修改就是给平衡树的一个区间 cover x， cover y。。因此可以直接平衡树维护折线，每次推折线的时候会加入一些点，这个可以用线段树维护。

如果遇到插入操作会破坏性质，所以可以二进制分组，时间复杂度 $O(n\log^2 n)$。

P8524

https://www.luogu.com.cn/problem/P8524

P5066

https://www.luogu.com.cn/problem/P5066

bzoj 3946

https://darkbzoj.cc/problem/3946

Top Tree