区间数据结构2

- 给定序列，元素有正有负，求有多少个子区间，区间和在 $[L, R]$ 范围内。

- 前缀和。
- $L \leq S_r - S_{l-1} \leq R$
- $L + S_{l-1} \leq S_r \leq R + S_{l-1}$
- 对每个 S_{l-1} ，统计合法的 S_r 数量。

- bzoj4627

- 二维平面有一些点，求一个边长为 R 的正方形最多能覆盖几个点。
- 坐标为 $[1,5000]$ 内的整数。

- 二维前缀和， $S_{i,j}$ 表示横坐标小于等于 i ， 纵坐标小于等于 j 的点数。
- $S_{i,j} = S_{i,j-1} + S_{i-1,j} - S_{i-1,j-1} + a_{i,j}$
- (x_1, y_1) 到 (x_2, y_2) 的矩形：
- $S_{x_2,y_2} - S_{x_1-1,y_2} - S_{x_2,y_1-1} + S_{x_1-1,y_1-1}$
- $O(5000^2)$

- bzoj1218

- 扫描线。
- 枚举纵坐标，设 f_i 表示横坐标在 $[i, i+R]$ 内的点数。
- 枚举纵坐标时相当于会加点和删点。
- 横坐标为 x 的点会影响 $x-R \leq i \leq x$ 内的 f_i 。
- 区间+-1，全局最大值。
- $O(nlogn)$

- 有 n 个一次函数， $f_i(x) = k_i x + b_i$ 。
- 两种操作：
- 1.修改第 i 个一次函数。
- 2.求第 l 个到第 r 个一次函数嵌套的结果，即 $f_r\left(f_{r-1} \dots f_l(x)\right)$ 。

- $f_i\left(f_j(x)\right) = f_i\left(k_j x + b_j\right) = k_i k_j x + k_i b_j + b_i$
- 每个区间维护区间的嵌套结果。
- 询问时取出定位的log个区间再合并一下。

- bzoj4499

- 给定一个序列，多次询问某个区间中有多少个不同的数。

- 枚举右端点 r 。
- 记 a_r 上一次出现为 a_{lst} ，则左端点为 $[lst+1,r]$ 多了一个不同的数。

- bzoj1878

- 有两种贝壳，火贝壳与光贝壳，火贝壳每次可以造成 x_i 点伤害，光贝壳可以造成 x_i 点伤害的同时，使得下一次伤害加倍。一次战斗中每个贝壳最多用一次。
- 插入删除【火/光】属性贝壳等操作，每次操作后，输出一次战斗总共能造成的伤害最大值。

- 给伤害较高的贝壳更优。
- 设光贝壳有k个，答案就是总和+前k大?
- 前k大都是光贝壳：去掉第k大，加上第k+1大。

- cf1398E

- 二维平面，支持三种操作，第一种操作是添加一个点(x,y),第二种操作是删除一个点(x,y), 第三种操作是查询严格在点(x,y)右上角的点中，横坐标最小的点，如果有多个点，选择纵坐标最小的那个。

- 二维?
- 按横坐标建线段树，每个叶子放一个set表示该横坐标对应的所有纵坐标。
- 每个线段树节点记该横坐标区间内的纵坐标最大值。
- 线段树上二分出最小的合法横坐标。
- 在对应的叶子set中upper_bound最小的合法纵坐标。
- $O(nlogn)$

- cf19D

- 一个长为 n 的区间，初始每个位置都为空。
- 每次会给出一个 x_i ，你需要找到一个连续的长度为 x_i 的空的区间并将其覆盖，若有多个满足，则覆盖最左边的。

- 类似最大子段和。
- 最长空的前缀，最长空的后缀，最长空的子段。
- 类似线段树二分找到最靠左的合法位置。

- bzoj1593

- 一个排列，判断是否存在长度不少于3的等差子序列。

- 只需判断是否有长度为3的即可。
- 考虑 a_i ，对于所有下标小于 i 的数字，若 $a_i - k$ 出现了，则 $a_i + k$ 必须出现，若 $a_i - k$ 没有出现，则 $a_i + k$ 必须不出现。
- 单点修改，区间查询字符串哈希值。

- bzoj2124

- 给定一个序列，每次把i到n中小于等于 a_i 的数排序并放回原位置，求每一操作后的逆序对个数。

- 把逆序对的贡献放在较大的那个数上。
- 一次排序，被排序的数的贡献都被清零。
- 每个数只被清零一次，即清零后赋一个INF，不再找到它。

- bzoj3333

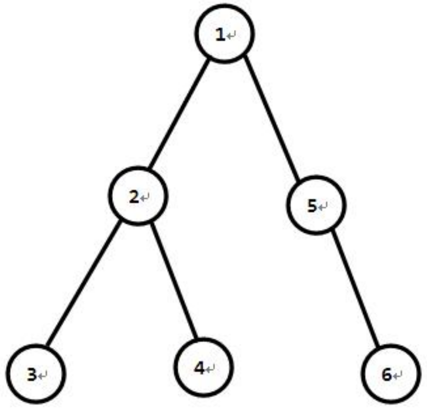
- 一个序列，多次询问一个区间的mex。
- mex：最小的没有出现的自然数。

- 枚举右端点。
- 权值线段树，记录权值为 i 的数最后一次出现的位置。
- 询问即找到最小的权值，其出现位置小于左端点。
- 线段树上二分，每个线段树节点记区间最小值。

- bzoj3585

- 一棵有根树，三种操作：
- 1.单点点权加
- 2.子树点权加
- 3.询问某节点到根路径的点权和

- 各种树上数据结构可做。
- 考虑欧拉序。
- 1 2 3 -3 4 -4 -2 5 6 -6 -5 -1



- 某个点的子树是它两次出现之间的所有位置。
- 到根的路径?
- 考虑到它的前缀，不在路径上的点会in一次out一次，in带一个+1的权，out带一个-1的权，可以抵消。
- 线段树修改时需要直到区间内有多少个-1，多少个+1。

- bzoj4034

- 一棵有根树，每个点有一个正的点权。
- 选择 k 条从叶子到根的路径，路径并的权值和最大。

- 贪心取最大路径，然后路径上的点权变为0，然后继续贪心。
- 删去一个点，相当于影响它子树内的点到根路径和。
- 每个点只会被删一次。

- bzoj3252

- 一棵树，有黑点白点。
- 操作会将某个点反色。
- 每次操作后输出距离最远的两个黑点的距离。

- 考虑用欧拉序维护两点距离。
- 设 dep_u 表示 u 的深度。
- $dis(u,v) = dep_u + dep_v - 2 \times dep_{lca}$
- 在欧拉序中， lca 是 u,v 表示的区间中深度最小的。
- 因此变为了 $dep_u + dep_v - 2 \times \min\{dep_x\}, u \le x \le v$ 。
- 存最大的 $dep, -2 \times dep, dep_x - 2 \times dep_y, -2 \times dep_x + dep_y$ 。
- 区间的答案就是左边最大的 dep ，右边最大的 $-2 \times dep_x + dep_y$ ，或者左边最大的 $dep_x - 2 \times dep_y$ ，右边最大的 dep 。

- bzoj1095