区间最值

区间最值问题

- $f[t][i]$ 表示 a_i 到 a_{i+2^t-1} 的最小值。
- $f[t][i] = \min(f[t-1][i], f[t-1][i+2^{t-1}])$
- 询问?
- $\min\left(f[k][l], f[k][r-2^k+1]\right)$
- 多次计算不影响最小值

- 给定两个长为 n 的序列 A, B ，求有多少个区间 $[L, R]$ 使得 $\max\{A_L, A_{L+1}, \dots, A_R\} = \min\{B_L, B_{L+1}, \dots, B_R\}$ 。
- $n \le 2 \times 10^5$

- 确定左端点 L 后，随着右端点 R 的增加，区间最大值单调不降，区间最小值单调不升，二分出使它们相等的右端点。

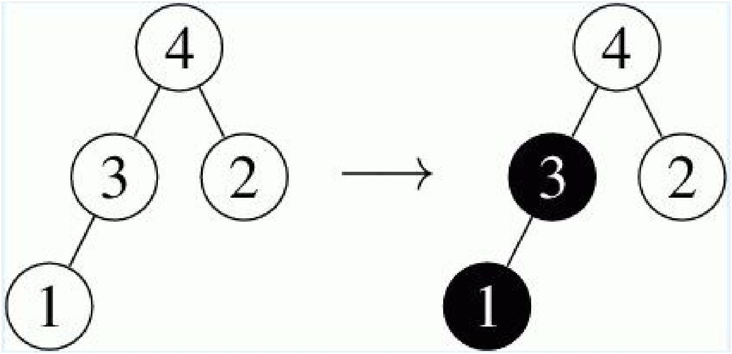
- cf689D

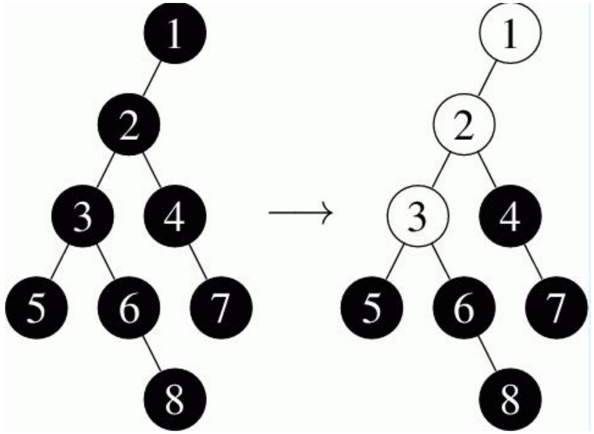
- 一张有向带权图，求一条从1号点出发的路径（可经过重复点和重复边），要求路径边权和至少为m，且边数最少。
- $n \leq 100$

- 倍增floyd。
- $f[t][i][j]$ 表示从 i 号点，走 2^t 条边到达 j 号点，边权的最大和。
- $f[t][i][j] = \max_{1\le k\le n}\{f[t-1][i][k] + f[t-1][k][j]\}$
- 按二进制位从高到低贪心。

- bzoj2165

- 往一棵有根树上放球，每个节点最多存一个球，若有子树节点为空，选择编号最小的那个子树节点对应的子节点继续下落。
- 从一个节点拿走球会导致它上方的一些球掉下来。
- 多次操作，有两种：
- 1. 在根节点上放若干个球，然后你需要输出最后一个球被放到了哪个节点上。
- 2.取走某个节点上的球，然后你需要输出有多少个球会因此掉落。
- $n, q \leq 10^5$





- 按子树最小编号顺序进行dfs，记录后序遍历。
- 放球就是按照这个顺序。
- 若一个节点被放球了，其子树内的所有节点也会被放球。
- 拿走一个节点上的球，相当于找到其最浅的有球的祖先，拿走它的球。
- 用堆维护空的节点，比较顺序按后序遍历中的顺序。
- 倍增找最浅的有球祖先。
- 从 i 号点向上跳 2^t 步的祖先：
- $fa[t][i] = fa[t-1]\left[fa[t-1][i]\right]$

- bzoj3133

- 一个 $n \times m$ 的矩阵，有一些格子 (x_i, y_i) 会在 t_i 的时刻被染成黑色。
- 求一个最早的时间点，使得矩阵中存在一个 $k \times k$ 的黑色子矩阵。
- $n, m \leq 1000$

- 对于每个 $k \times k$ 的子矩阵，它被全部染黑的最早时间点就是内部每个格子染色时间的最大值。
- $f[t][i][j]$ 表示以 (i, j) 为左上角的 $2^t \times 2^t$ 子矩阵中的最大值。
- $f[t][i][j] = \max(f[t-1][i][j], f[t-1][i+2^t][j], f[t-1][i][j+2^t], f[t-1][i+2^t][j+2^t])$
- 每次询问就是四个角的 f 取最大。

- cf846D

- 有一个长为 n 的序列 $A_0, A_1, \dots, A_{n-1}$ ，每次操作之后会变成一个新的序列 B ，其中 $B_i = \gcd\left(A_i, A_{(i+1)\ mod\ n}\right)$ 。
- 求在第几次操作之后，序列中的所有元素都相等。
- $n \le 2 \times 10^5$

- 全部相等就是全部变成了所有数的gcd。
- 经过 k 次操作之后， A_i 就变成了 $\gcd\left(A_i^0, A_{i+1}^0, \ldots, A_{(i+k)\ mod\ n}^0\right)$ ，其中 A^0 为原始序列。
- $f[t][i]$ 表示 $\gcd\left(A_i^0, A_{i+1}^0, \ldots, A_{(i+2^t)\ mod\ n}^0\right)$ 。
- cf1547F

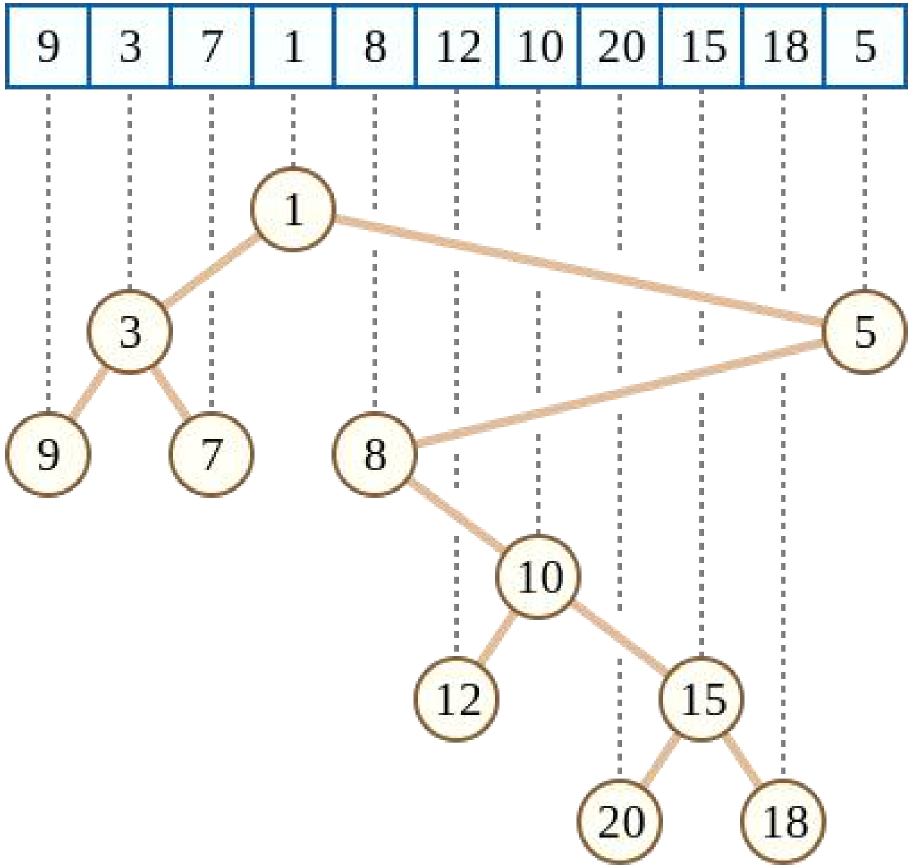
- 一个长为 n 的排列，只知道所有 $n-1$ 组相邻元素的大小关系，构造出满足条件的LIS最长和LIS最短的排列。
- $n \le 10^5$

- 7
- >><>><
- 5 4 3 7 2 1 6
- 4 3 1 7 5 2 6

- 最短LIS就是最长的连续<加一。
- 从后往前考虑每一个<的连续段，用一段还未使用的连续数字填。
- 最长LIS就是<的总数加一。
- 从后往前考虑每一个<，尽量填较大的数。

- cf1304D

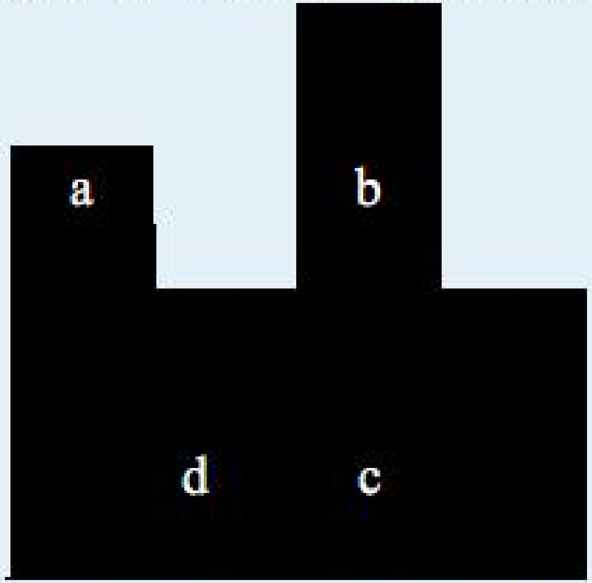
笛卡尔树



求法?

- 每次找区间最值，然后分治。 $O(nlogn)$ 或 $O(n)$ 。
- 维护单调栈，栈中元素是当前笛卡尔树的右链。 $O(n)$

有 N 个 $1\times h[i]$的矩形小棋盘，底边边长为 1，在一条直线上拼成了一个畸形的棋盘。高度 $h[i]$给出，第 i 个矩形的高为 $h[\mathrm{i}]$，例如 $h=[3, 2, 4, 2]$的图形如下：



若两个车相互攻击仅当它们在同一列，或在同一行且在这一行它们之间棋盘格子都是存在的，例如上图中 a 与 b 是相互不攻击的，c 与 d，b 与 c 均为相互攻击的。

现在要在这棋盘上放置恰好 K 个相互不攻击的车，问有多少种方案。

- $n \le 500$

- 找到最小的 h_i 之后，两边的车若放在高于 h_i 的地方，是互不影响的。
- 建立笛卡尔树， $f[i][j]$ 表示在 i 的子树内，放 j 个车，并且放的位置都高于 h_{fa_i} 的方案数。
- 两个儿子直接合并。
- 考虑要在 h_i 到 $h_{fa_i}+1$ 的高度内放若干个车。
- 若放 k 个车，那么就是在 h_i 到 $h_{fa_i}+1$ 内选出 k 行，在可用的列中选出 k 列，然后 $k!$ 组合一下。

- bzoj2616

- 给定两个长为 n 的序列 c, d 和一个常量 a，求一个区间 (l, r)，最大化 $f(l,r) = (r - l + 1) \times a - \sum_{i=l}^{r} c_i - \max_{l<i\le r}(d_i - d_{i-1})^2$。
- $n \le 3 \times 10^5$

- 先找到最大的 $d_i - d_{i-1}$ ，考虑跨过它的区间。
- 要考虑的部分相当于 $a \times r + sum[r] - a \times (l-1) - sum[l-1]$ 。
- 右半边取最大，左半边取最小。
- 类似笛卡尔树分治即可。

- cf1107G

- 给定一个排列，对于一个连续的区间 $a_l, \ldots, a_r$ ，若 a_l 和 a_r 都是这个区间的最值，就在 (l, r) 之间连边。求1到n的最短路。
- $n \leq 5 \times 10^5$

- 从1到n一定无法绕过最大值的位置。
- 将序列划分为两个部分，每个部分递归找最小值，再继续递归找最大值…

- cf1696D

- 一个长为 n 的序列 a ，保证 n 为2的幂次，每秒钟有一个操作，对于所有数字同时： $a_i = a_i \ xor \ a_{(i+1) \ mod \ n}$ 。
- 求最早的使得所有数字变为0的时刻。
- $n \le 2^{20}$

- 一旦某次操作后全部变为0，后续的时间一直全为0。
- 第 2^k 次操作会使第 i 位变成 $a_i\ xor\ a_{(i+2^k)\ mod\ n}$。
- 倍增， 2^k 步之后的局面容易计算，若未全0则加入。
- 可以保证解存在，因为 2^n 步之后就是 $a_i\ xor\ a_i$ 。

- cf1848F

- 两个长为 n 的序列 a, b ，每次询问一个 L, R, m ，你需要求出一个 $L \le l \le r \le R$ ，使得 $\sum_{i=l}^{r} a_i \ge m$ ，同时 $\max(b_l, \ldots, b_r)$ 尽可能小。

- $n \le 10^5$

- 若 $L < l \leq r < R$ ，则 $[l,r]$ 一定是以某个 b_i 为最大值的最大能扩展的区间，这样的区间只有 n 个。
- 否则，一定有 $L = l$ 或 $R = r$ ，二分另一个端点即可。

- 有 n 个人，每个人有 p_i, m_i ，你需要让所有人为你投票。
- 第 i 个人为你投票，你要么需要给他 p_i 的钱，要么已经有其他 m_i 个人为你投票。
- 求最小花费。
- $n \leq 10^5$

- 考虑一个投票序列，对于第 i 个投票的人，若他的 m 不超过 $i-1$，则不需要产生花费。
- 相当于是省尽量多的钱。
- 枚举每个位置 i，大根堆存放 $m < i$ 的人，取出最大值减掉。

- cf1251E

- 给定一个序列，一个区间 (l,r) 有价值 $f(l,r)$ 。
- 给定一个 k ，将序列恰好分为 k 段，使得价值最大的段的价值尽可能小。
- 这里设 $f(l,r)$ 是一个可以 $O(1)$ 计算的一个函数，且 l 固定时关于 r 单调， r 固定时关于 l 单调。

- 最大值最小，显然想到二分答案 x 。
- 从1开始，对于当前左端点找最大右端点使得 f 不超过 x ，分段，然后左端点跳到右端点+1，最终比较段数和 k 来二分即可。
- 如何找到最大右端点?
- 第一段以1为左端点，枚举右端点。
- 后续每段以上一段右端点+1为左端点，枚举右端点。

- 给定一个序列，一个区间 (l,r) 有价值 $f(l,r)$ 。
- 给定一个 k ，将序列恰好分为 k 段，使得价值最大的段的价值尽可能小。
- 这里设 $f(l,r)$ 是一个只能用 $O(r-l+1)$ 复杂度计算的一个函数，且 l 固定时关于 r 单调， r 固定时关于 l 单调。

- 最大值最小，显然想到二分答案 x 。
- 从1开始，对于当前左端点找最大右端点使得 f 不超过 x ，分段，然后左端点跳到右端点+1，最终比较段数和 k 来二分即可。
- 如何找到最大右端点?
- 虽然左端点固定时 f 关于右端点单调，但二分右端点的复杂度不对，因为二分过程中可能会超过最终右端点太多。
- 考虑倍增，先从小到大枚举 t ，找到最大的 t 使得 $f(l, l+2^t-1) \leq x$ 。
- 接下来再从 $t-1$ 开始往小枚举 2^i 倍增。
- 只会计算 f 共log次，且每次计算的长度都不超过最终段长的两倍。