Solution

清华大学 陆宏

游戏 (game)

- 有一棵n个点的树，第i个点上有a_i块石头。
- Alice和Bob要用这棵树玩一个游戏，Alice先选取一个点作为起始点，在该点上放置一枚棋子，Alice先手，两人轮流进行以下操作：
  - 取走当前棋子所在点的一块石头，然后把棋子移到相邻的点上。如果当前棋子所在点上没有石头，那么进行当前这步操作的人就会因为无法取走一块石头而输掉游戏。
- 求Alice有多少选取起始点的方式使得他能赢得这场游戏。
- $n \le 2 \times 10^5$

游戏 (game)

- 若对于一个点，与其相邻的节点上的石子数量都不少于它，那么这个点是必败态。因为无论先手怎么移动，后手总可以移回来。
- 由于这是一棵树，若先手将棋子往石子数不小于当前位置石子数的点移，后手一定可以直接将棋子移回去，从而使先手不仅无法移动至该节点，还会陷入更劣势的局面（可选择的余地变小）。
- 故每次操作的人只会将棋子往石子数比当前位置石子数小的点移。
- 于是直接按石子数从小到大，确定每个点是必败态还是必胜态。
- 时间复杂度$O(n)$。

跳 (jump)

- 有一辆汽车，油箱初始上限为v，有n个加油站在一个数轴上，第i个加油站的位置为x_i。你想要访问过每一个加油站，有两种方式进行移动：
  - 从当前所在加油站移动到与当前加油站距离不超过油箱上限的加油站。
  - 如果油箱上限不为0，那么可以从当前所在加油站移动到任意一个加油站，然后油箱上限减半（下取整）。
- 每个加油站可以被多次访问。判断是否可以从每个加油站出发，访问所有加油站。$2 \le n, v \le 2 \times 10^5$

跳 (jump)

- 注意到2操作最多进行$\log v$次，且一定会把2操作尽可能用完。
- 故题目等价于，给定一个集合S，把序列分成$|S|$段，使得每一段都可以和集合里的一个数匹配并且这段相邻加油站的距离最大值比这个数小。
- $|S|$就是通过2操作移动的次数，S中的数就是经过每次2操作之后的油箱容量（以及初始油箱容量）。

跳 (jump)

- $f(S)$表示用了S集合中数字，从左往右最远能访问到哪个加油站。
- 转移枚举当前使用的数是啥，通过预处理从每个位置开始，使用每个数字往右走最远能走到哪个位置即可实现快速转移。
- 同理令$g(S)$表示用了S集合中数字，从右往左最远能访问到哪个加油站。

跳 (jump)

- 虽然看似有n次询问，但注意到，对于从每个加油站出发，仅使用1操作能走到的加油站，它们的答案一定相同。这样划分出来的序列一定不超过$\log v$段，否则答案全是Impossible，即相当于只需要处理$\log v$次询问。
- 对于每次询问，枚举左边用了S集合的数字，那么根据$f(S)$和$g(\bar{S})$即可判断是否能走到所有加油站。
- 时间复杂度$O((v+n)\log v)$。

排列 (permutation)

- 你有一个长度为n的排列。定义一个区间$[l,r]$是好的，当且仅当把这个区间的数按升序排序后是连续的（相邻两个数之差为1）。
- 有q个询问，每次询问一个区间$[l_i,r_i]$，问有多少个好的区间$[x,y]$，满足$l_i \le x \le y \le r_i$。
- $1 \le n,q \le 2 \times 10^5$

排列 (permutation)

- 先考虑如何判断一个区间是否是好的。若区间最大值−区间最小值=区间右端点−区间左端点（$\max-\min=r-l$），则该区间是好的。
- 给定q个区间，如何判断这些区间是否是好的?
- 将询问按右端点排序，枚举右端点，线段树第i个叶子节点维护以i为左端点到目前枚举的右端点的区间的$(\max-\min)-(r-l)$的最小值及其出现次数，当$(\max-\min)-(r-l)=0$时，区间最小值出现次数即为所求。

排列 (permutation)

- 但是题目不是判断一个区间是否是好的，而是求一个区间中有多少个子区间是好的。那么线段树的节点不仅要维护当前最小值和最小值出现次数，还有维护历史最小值为0的出现次数。
- 也就是说，每次考虑完一个右端点，需要将当前枚举的右端点状态下的线段树上的0的个数计入到线段树上每个节点的历史最小值个数。

排列 (permutation)

- 先考虑暴力处理，如果递归到某个点，它的最小值不为0，那么就停止递归它的子树。否则就把当前点的最小值出现次数加到历史最小值出现次数中。
- 但是为了保证复杂度，线段树不能花费额外代价去处理这件事，即必须打一个懒标记来维护这个区间的历史最小值出现次数，然后再考虑标记下放的问题。

排列 (permutation)

- 这个标记并不总是能下放，如果子节点在打上标记的那个时候的最小值为0，这个标记才能下放到该子节点上。
- 但是，线段树并不能记录每个标记被打上的时间，因为这么做一个点可能存在很多标记；也不能通过现在的最小值来判断标记是否能下放，因为这个区间的最小值可能会被修改（被修改时不一定会递归到这个区间，如父节点打区间加标记）。
- 故希望得到一个快速判断打标记时子节点最小值是否为0的方法。

排列 (permutation)

- 注意到，当一个节点有这个标记时，这个节点当时的最小值一定为0。
- 根节点的最小值一定为0，如果一个节点的最小值和其子节点的最小值一样，并且这个节点有标记，那么该子节点当时的最小值也一定是0，即标记可以下传至该子节点；否则不应下传标记。（这个节点和其子节点的最小值相同，意味着它们在当时的最小值也相同，均为0。如果存在针对该子节点的区间加操作，标记在这个操作执行时就应被下放）

排列 (permutation)

- $(\max - \min) - (r - l)$的更新可以使用单调栈处理，均摊$O(n)$次区间修改操作。
- 线段树每个节点维护$(\max - \min) - (r - l)$的最小值，最小值出现次数，区间加懒标记，历史最小值出现次数，历史最小值出现次数的懒标记。
- 对询问$[l, r]$，当右端点枚举到r时，查询$[l, r]$区间的历史最小值出现次数和。
- 时间复杂度$O((n + q)\log n)$。