

Arc4hv - 题解

k 较小

随便怎么搞都可以（比如预处理所有可能的询问的答案），期望得分 20。

q 较小

考虑暴力。

首先枚举赢的比赛轮数 cnt ，然后枚举 X 在第 cnt 轮比赛的哪一场，则 X 必然是一个形如 $[p, p + 2^{cnt} - 1]$ 区间里最大的数。

直接考虑能不能通过交换将 X 换进这个区间和将所有比 X 大的数换出区间即可。

时间复杂度 $O(qn \log n)$ ，期望得分 30。

$Y = 1$

考虑如何判断一个形如 $[p, p + 2^{cnt} - 1]$ 的区间是否合法。

对于 $Y = 1$ 的情况，合法的条件就是这个区间的次大值 $\leq X$ （因为若 X 在区间中，则最大值能够被换出去，否则可以交换 X 和这个区间的最大值）。

预处理所有区间的次大值即可，预处理的时间复杂度为 $O(n \log n)$ ，可能需要压一下空间。

期望得分 20，结合上文算法可过 50。

离线

考虑将所有询问按照 X 递增排序。

对于一个询问 (X, Y) ，我们将所有小于 X 的元素都插入到序列里。

现在考虑如何判断一个形如 $[p, p + 2^{cnt} - 1]$ 的区间是否合法。

若这个区间被插入了 P 个元素，那么分两种情况讨论：

1. X 在这个区间中，则只需要要求 $Y \geq 2^{cnt} - P - 1$ 即可。（这相当于将所有大于 X 的元素换出去，注意若 $X < 2^{cnt}$ ，我们永远不可能让 X 成为这个区间的最大值，因此此时一定不合法）
2. X 不在这个区间中，则只需要要求 $Y \geq 2^{cnt} - P$ 且 $Y > 0$ 即可。（这相当于将 X 与这个区间中一个大于它的元素交换，然后将所有其他大于 X 的元素换出去。同理若 $X < 2^{cnt}$ 不合法）

观察发现我们只需要维护两种东西，第一是所有长度为 2^{cnt} 的区间中 P 的最大值，第二是每一个区间被插入了多少个元素。

离线之后这两样东西都很好处理。

时间复杂度为 $O(n \log n)$ 。

正解

在线的问题在于，很小的空间限制导致我们不能用诸如可持久化数组一类的数据结构维护每一个 X 的最大值和插入的元素个数。

还是考虑分类讨论，我们讨论 X 是选择留在自己所在的区间里还是换到另一个区间去。

留在自己所在的区间里，意味着这个区间只能有 Y 个大于它的数，这个东西看起来不好维护，我们先无视它。（强行维护的话，你需要 $O(n \log n)$ 的空间，这是完全无法接受的）

先考虑换到另一个区间里去怎么做。

对于每种 cnt ，我们同时考虑所有长度为 2^{cnt} 的区间。贪心地想，若我们有 Y 次交换机会，我们肯定会选择这些区间中第 $Y + 1$ 大的数最小的那个，然后将这个区间里前 Y 大的数都换掉，因为这样可以尽量让 X 大于其它的数。

我们现在要求的就是：所有长度为 2^{cnt} 的区间里，第 K 大的数最小是多少。

这可以通过预处理处理出来，直接从小到大插入所有数就行了。同时我们观察到， $\sum 2^{cnt} = 2n - 1$ ，因此空间复杂度是 $O(n)$ 的。

有了上面的一个辅助数组，我们也能处理 X 留在自己所在区间里的情况了。

考虑我们在预处理（从小到大插入数）的过程中，在我们插入 X 的时候，对于某个特定的 cnt ，若在插入 X 前时 X 所在区间中数的数量不是所有长度为 2^{cnt} 的区间中数的数量最多的，则我们可以断言此时留在自己区间是不优秀的。因为此时若想让 X 在自身所在的区间中达到最大，我们至少还需要执行 $2^{cnt} - \text{此时区间中数个数}$ 次交换，此时将 X 换到其他区间里所需的交换次数不会多于上述次数。唯一的例外就是插入 X 的时候区间满了，则此时是留在自身所在的区间最优，需要特判处理。

我们现在知道了对于哪些 cnt ， X 留在自身区间里可能是优秀的，那么如何判断 Y 次是否够呢？直接计算显然是不可行的。考虑上面预处理出的数组。由于我们知道当 X 被插入所在区间后，它所对应的区间一定是所有区间里数的数量最多的，则此时 X 一定会更新我们上面提到的 "所有长度为 2^{cnt} 的区间里第 K 大的数最小是多少" 的数组，我们只需要检查在上面提到的数组里是否满足第 $2^{cnt} - Y - 1$ 大的元素小于 X 即可。具体细节见标程。

时间复杂度 $O(q \log n)$ ，空间复杂度 $O(n)$ 。