可爱的背包

正解

将所有物品按照 r_i 从小到大排序，考虑 dp，令 f_i 表示当前加入物品中最靠右的右端点为 i 的最大价值，按照从左往右的顺序依次将物品放入背包，那么每次加入一个物品时进行转移：

$$f_{r_i} = \max_{j<l_i} f_j + v_i$$

维护 dp 数组的前缀 max ，排序使用桶排即可做到线性。当然带一个 log 也能过。

加边

正解

对于一次询问 x，保留所有宽度 $\geq x$ 的边，如果此时图仍然连通则无需加边，否则需要加边让图重新连通。可以发现，代价最小的加边方案是找到全局 a_i 最小的点 p 所在的连通块，从其他每个连通块中 a_i 最小的点向点 p 连边，若 S 为连通块集合则此时总代价为

$$(|S|-2)\times a_p+(\sum_{T\in S}\min_{x\in T}a_x)$$

将询问离线按 x 从大到小排序，每次加入所有宽度 $\geq x$ 的边，用并查集维护连通块数量和每个连通块中的最小 a_i。假设 n, m, q 同级，时间复杂度 $O(n\log n)$。

奇怪的操作

正解

由于每次只修改一个位置且没有后效性，所以可以考虑先求出不变的答案，然后算上位置改变的贡献。

对于每个 i，求出 v_i 表示前 i 个数中第 b_i 大的值，v'_i 表示第 b_i+1 大的值（若不存在则为 0），c_i 表示前 i 个数中前 b_i 大的数之和。这些值可以从前到后把序列扫一遍的同时用权值线段树查询。

先求出初始的 $\sum c_i$。对于一次询问将 a_x 改为 k：

1. $a_x < k$：对于每个 $i \ge x$，

- 若 $v_i \le a_x$，说明原先 a_x 是前 b_i 大的值，c_i 要先减去 a_x。
- 若 $v_i \le k$，说明改成 k 后是前 b_i 大的值，c_i 要再加上 k。
- 若 $a_x < v_i \le k$，说明原先 a_x 不是前 b_i 大，改完了是，c_i 要减去 v_i。

2. $a_x > k$：对于每个 $i \ge x$，

- 若 $v'_i < a_x$，说明原先 a_x 是前 b_i 大的值，c_i 要先减去 a_x。
- 若 $v'_i < k$，说明改成 k 后是前 b_i 大的值，c_i 要再加上 k。
- 若 $k \le v'_i < a_x$，说明原先 a_x 是前 b_i 大，改完了不是，c_i 要加上 v'_i。

将询问离线按 x 从大到小扫描线，用两棵权值线段树分别维护 v_i 和 v'_i，支持查询值域区间内有多少个 v_i 以及 v_i 的和即可回答询问。

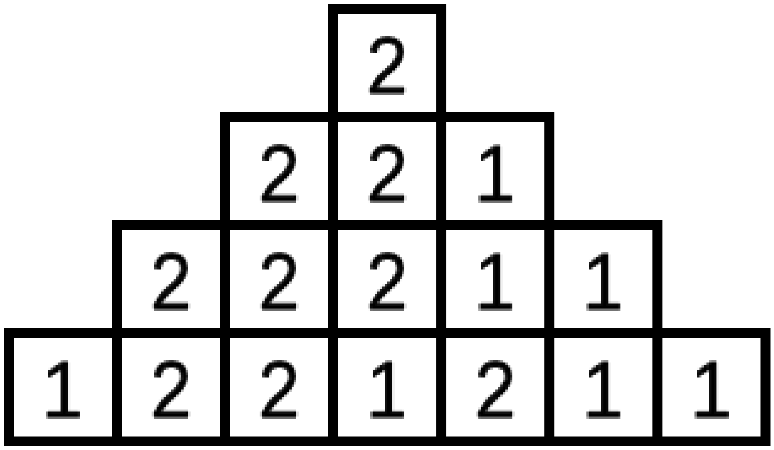
时间复杂度 $O(n \log n)$。

函数

算法 1

对于 $q=1, a_i \le 2$ 的部分。

将整个序列的变换看成一个三角形画出来如下：



显然可以发现如果出现一对相邻位置相同，那么这个位置会一直向上传递直到出现一个更靠近中间的相邻相同的位置然后被干掉。

于是结论就是答案是最靠近中间的一个相邻相同的位置的值，而由于区间长度是奇数，可以发现不可能出现矛盾。如果不存在相邻位置相同，那么答案显然是与中间位置相反的值，时间复杂度 $\mathrm{O}(n)$。

算法 2

对于 $q=1$ 的部分。

二分，将小于二分值的看作 1，大于等于二分值的看作 2，即可像算法 1 一样解决。

时间复杂度 $\mathrm{O}(n\log n)$，结合暴力期望得分 50 分。

算法 3

考虑离线后整体二分，维护序列上相邻相同的位置，那么每次的 check 即转化为求所有小于 x 的最大的值和大于 x 的最小值，可直接使用平衡树维护。

时间复杂度 $\mathrm{O}(n\log^2 n)$，期望得分 100 分。