A.高爸、O(1)算法与乡下人

30 pts

按题意模拟即可，注意取模。

94 pts

注意另外的 64 分保证了 $typ = 1$ ，那么我们只要关注 n 这个尼最后一次被扶贫操作即可。

事实上我们可以把操作倒过来做，用并查集简单维护即可回答每个尼最后的存款。

100 pts

容易发现在一次扶贫操作中，对于存款相同的尼哥，他们的感恩值贡献是一样的，所以我们可以把某些存款相同的尼哥放在一起处理。

又注意到我们的修改本质上是区间覆盖，所以引出了一个经典做法：对于一段存款相同的尼，我们可以当做是一个尼放一起处理，于是当前的尼哥序列就被我们变成了许多个连续的小段，每个连续段的尼存款都相同。

考虑处理一个修改，由于可能存在连续段一部分在修改内部，一部分在修改外部，所以我们先把这些连续段手动切开，注意到这样的连续段最多出现两个，也就是左端点和右端点出现两个跨越端点的连续段，我们至多进行 $O(1)$ 次切开连续段的操作，连续段个数至多增加 $O(1)$ 个，那么经过所有操作过后，连续段的总个数不超过 $O(m)$。

那如何计算感恩值呢？我们找到被 $[l,r]$ 包含的那些连续段，对于一整个连续段可以一起处理，因为在经过这次修改过后， $[l,r]$ 这一段的所有尼哥存款相同可以缩成一个连续段了，所以这些被包含的连续段在计算完过后就可以被删除，注意到对于一个连续段，删除前只会被至多计算一次，所以只要实现得当，计算次数就是 $O(m)$ 的。

只要使用题目中提到的 $O(1)$ 算法 `set < pair <int, int> >` 维护一下连续段即可。（当然 `set < pair <int, int> >` 并不是真的是 $O(1)$ 的，而是 $O(\log n)$ 的，而且由于内部实现为红黑树，所以常数比 `priority_queue` 大得多，所以请大家努力成为高爸）

题外话

原本 f 函数想给一个关于 x 的多项式比如 $f(x) = x^2$ 之类的，一想到有些选手会熟练的用二项式定理展开之后之后用线段树维护，这是我最不愿意看到的，于是我随便从网上找了个哈希用的 xorshift 调了调参，改装成了现在的 f ，不然这一场线段树就太多了。

B.ry泡妹子

99pts

$dp_{i,j,k}$表示前i个，选了j个，最后一个的权值为k的最大快乐值和方案数

转移肯定是一个$dp_{i,j,k}=\max_{l<k}dp_{i-1,j-1,l}$的形式

前缀优化一下就做完了

100pts

我们拿树状数组维护第三维的前缀，做完了（

Hint: max是取值大的，值相等方案数相加

C.clb与别墅

Problem by xryjr233

96pts

考虑每一对数(u,v)，满足$u<v$,$a_u\geq a_v$的贡献。然后把序列倒过来再算一次得到答案。

那么它产生贡献的条件是：这两个数都在子序列中，而且两个数之间只出现了$<a_v$的数。

我们记这两个数之间$<a_v$的数数量为k，那么这对数的贡献是$2^{n-v+u-1+k}$。

于是枚举v，从后往前枚举u，同时维护k即可，时间复杂度$O(n^2)$。

100pts

~~已经不是普及组难度了~~

依然只考虑$u<v$,$a_u\geq a_v$的。

考虑从小到大枚举数值，考虑等于这个数值的位置p。

记前i个数里$<a_p$的个数为s_i，那么位置p的贡献是

$$\sum_{q=1}^{p-1}[a_q\geq a_p]2^{n-p+q-1}\times 2^{s_p-s_q}$$

即

$$2^{n-p+s_p}\sum_{q=1}^{p-1}[a_q\geq a_p]2^{q-1-s_q}$$

后半部分的和可以简单用线段树维护。

时间复杂度$O(n\log n)$。

D.颜色对了

考虑建图，如果两点的切比雪夫距离大于 K 就连一条边。那么每个连通块必须是二分图，方案数是 2 的连通块个数次方。

直接做是 $O(n^2)$ 的，只有 50 分。考虑优化建图。

首先两维的图可以分开建，只要考虑一维情况。相当于数轴上有若干个点，若两点距离大于 K 就连边。发现只要考虑最左、最右两点连出的边，图的联通情况就和原图一样了。于是时间复杂度优化成了 $O(n)$，可以获得 100 分。