中位数

不难发现，对于一个数而言，只有当与它相邻的两个数都与它不同时，一次平滑操作才会使这个数改变。

于是多次平滑操作会影响的只有 0 和 1 交替出现的连续段，并且每执行一次平滑操作会使连续段的长度减少 2，我们可以据此很容易地得到所需的次数和最终的序列。

时间复杂度 $O(n)$ 。

千帆渡

设 f[i][j]表示序列 a 前 i 个数中的某一个数与 b_j 作为公共递增子序列的末尾时子序列的最大长度，显然有如下转移：

1. 若 a_i=b_j，则 $f[i][j]=\max\{f[i-1][k]\}+1(k<j,b_k<a_i)$ ；

2. 若 a_i≠b_j，则 $f[i][j]=f[i-1][j]$ 。

不难发现，第一种转移可以在枚举 j 的时候记录一个最值来优化掉枚举 k 的复杂度。

转移的时候顺便记录一下方案即可，时间复杂度 $O(nm)$ 。

斐波那契

定义类 fibonacci 数列表示满足 fibonacci 数列递推式但前两项不一定都为 1 的数列。

考虑直接在线段树中记录区间加类 fibonacci 数列的标记。

我们在每个线段树节点记录加上的类 fibonacci 数列的前两项 c,d，那么区间[L,R]内任意一个位置 i 被加上的数都可以表示成 a_{i-L+1}×c+b_{i-L+1}×d 的形式，整个区间被加上的数的和也可以表示成 sa_{R-L+1}×c+sb_{R-L+1}×d 的形式，其中数列 a,b,sa,sb 都可以递推预处理出。

显然两个类 fibonacci 数列逐项相加得到的还是类 fibonacci 数列，因此在线段树标记合并时可以直接把前两项各自相加。

时间复杂度 $O((n+m)\log n)$ 。

~~当然也有模意义下强行算通项的做法，不过要麻烦得多。~~

电梯

一道相当有思维难度的好题，显然主要问题在于怎样缩小图的规模。

考虑建出 x 个点，标号 0~x-1。对于每一个点 i，将它向点(i+y)%x 连一条边权为 y 的边，向点(i+z)%x 连一条边权为 z 的边。然后以点 1%x 为起点求单源最短路，那么到每个点 i 的最短路 dis_i 即表示通过若干次+y 和+z 的操作所能到达的满足%x=i 的最小楼层，最后的答案即为 $\sum_{i=0,dis_i\le h}^{x-1}(\left\lfloor\frac{h-dis_i}{x}\right\rfloor+1)$ 。

直接用 Dijkstra+堆求最短路即可，时间复杂度 $O(x\log x)$ 。