考虑判断 (p,s) 是否合法。记 $x_i=p_i-p_{i-1},y_i=s_i-s_{i+1}$。能相对简单地看出的条件有：

- $s_1=p_n$
- $\forall 1\le i\le n, 0\le x_i,y_i\le 1$

记 $s_1=L$。再就是，对于 $x_i=y_i=1$ 的位置，必有 $p_i+s_i=L+1$。

还有，对于所有位置必须有 $p_i+s_i\ge L+1$。这是因为，我们取出整个序列的 LIS，若其包含 i 则显然成立，否则 a_i 一定能对 p_i,s_i 中的一个产生贡献。

上述条件是充要的，构造如下：

不存在 $x_i=y_i=1$ 时，先倒着扫 $x_i=1$ 的位置，从 n 开始往下填；然后正着扫 $y_i=1$ 的位置，从 1 开始往上填；再把剩余的数降序填入剩余位置。对于一般情况可以按照 $x_i=y_i=1$ 的位置分段，每段是独立的。

使用 dp 来计数。设

$$z_i=\sum_{j=i}^{n} y_j-x_j$$

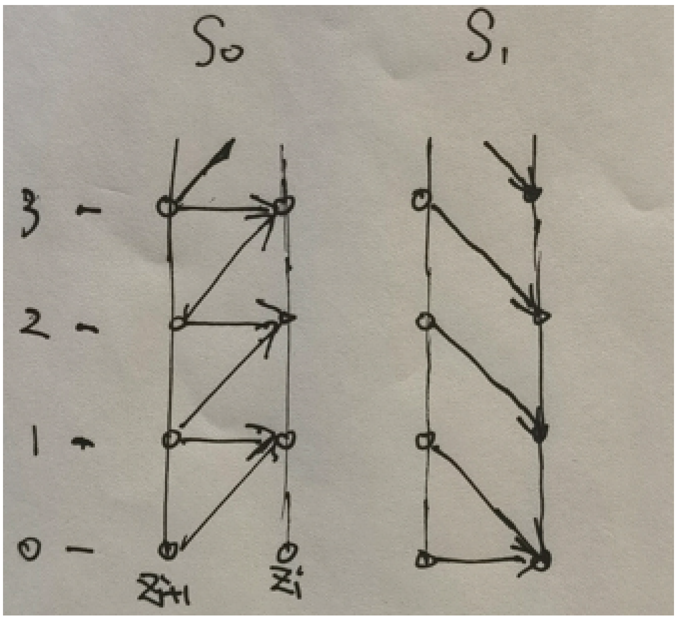
写一下式子。$p_i+s_i\ge L+1$ 等价于 $z_{i+1}+y_i\ge 1$。$x_i=y_i=1$ 的限制即为 $z_{i+1}=0$。

分析可得，z 合法当且仅当 $z_1=z_{n+1}=0$ 且 $(z_i,z_{i+1})\in S_{x_i}$，其中：

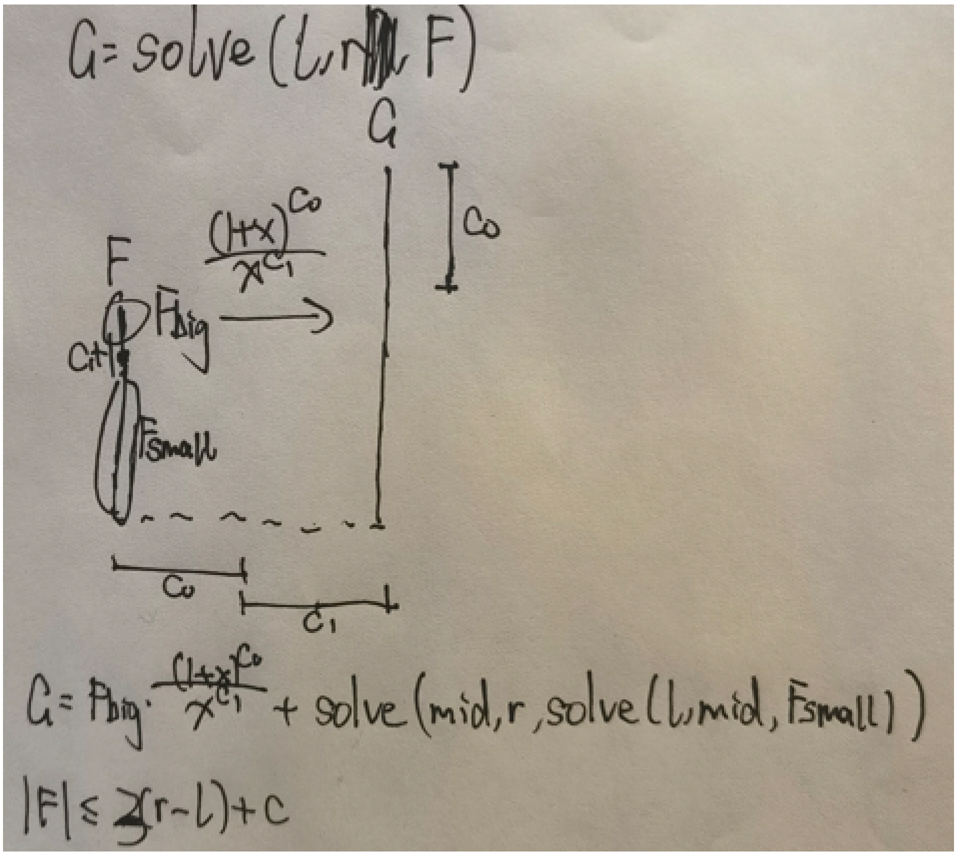
- $S_0=\{(u,u-v)|u\ge 1,0\le v\le 1\}$
- $S_1=\{(\max(0,u-1),u)|u\ge 0\}$

直接 dp 是 $O(n^2)$。

要做到更优复杂度，我们考虑分治。把 dp 状态看成格点，把相邻两列的转移画成下图：



定义 $solve(l,r,F)$：对于 $0\le i<up$，已知走到网格图 (l,i) 的方案数，生成函数为 F；函数返回走到 (r,i) 的生成函数。



按上图方式递归可以保证 $up=O(r-l)$，使用 ntt 进行卷积，时间复杂度 $O(n\log^2 n)$。

但是分治 ntt 超纲了，所以只给了 1 分。

实现时把各种上下界卡得紧一点，再预处理一下单位根，常数超小。