这是一套纯搬运的题目。

排列

原题链接

当$N=2$的时候显然无解，下面的构造给出了其他N的解。

注意到我们如果搞出一个排列$p_{k,i}=i+2^k\pmod N$意义下循环），那么$p_{k,i}^{-1}=i-2^k$，也就是说q_k可以理解为选择向左右平移2^k步。那么依次用$p_0\sim p_{d-1}$可以对任意的初始值以$-2^d+1\sim 2^d-1$中所有的奇数作为偏移量。

当N为奇数的时候，注意到对于任意两个数$x\le y$，偏移量可以是$y-x$或$y-x-n$，这两个一定一奇一偶，因此选择$p_0\sim p_{\lceil\log_2 N\rceil-1}$即可，共$\lceil\log_2 N\rceil$个。

当$N\bmod 4\equiv 0$的时候，用同样的方法会得到两个同奇偶的数。考虑稍微修改一下，先在前面补一个排列$q=(4k-1,4k,4k-2,4k-3)$。这样$\forall i$，q_i和q_i^{-1}中一奇一偶，因此对于一对x,y，可以先从x走到一个与y奇偶性相异的位置，就可以类似刚刚的做了。注意此时我们可以选择走一个绝对值较小的方向，只需要用到$p_0\sim p_{\lceil\log_2 N\rceil-2}$，仍然共$\lceil\log_2 N\rceil$个。

当$N\bmod 4\equiv 2$的时候，显然$N=2$无解，其他情况考虑再修改一下。刚刚$N\bmod 4\equiv 0$的做法问题在于最后会剩余两个数不能凑出长度为4的环，那么再在q前面补一个排列r，使得$r_1=N-1,r_2=1,r_{N-1}=N,r_N=2$，其他位置$r_i=i$，这样就可以保证每个位置经过$r$后都有办法不在$N-1$和$N$。后面类似$N\bmod 4\equiv 0$构造即可，共$\lceil\log_2 N\rceil+1$个。

时间复杂度为$\mathcal{O}(N\log N)$。

异或

原题链接

先特判掉$N \le 2$。

分析一下容易证明有解必须满足$S \ge X$且$S \equiv X(\mod 2)$。

注意到不考虑最小化最大值的要求，答案只跟每个位的1的个数有关。令dig_i表示从低往高第i个位1的个数，那么一个基本方案是$dig_i = [2^i]X + 2 * [2^i]((S-X)/2)$。可以证明其他所有方案都可以由这个基本方案每次将$i+1$位的2个1换成$i$位的4个1得到。注意到这个基本方案每一位不超过3，因此$N \ge 3$的时候必定有解。

现在考虑最小化最大值。显然可以考虑按位二分答案，对于一个上界M，我们要判定是否有一个dig的方案，使得存在一种分配方案令N个数都不超过M。对于一个固定的dig（$dig_i \le N$），显然有一个按位贪心的过程：从高往低看每个位的数，记录确定小于M的数字个数K，如果M第i位是0，就要求必须$K \ge dig_i$，为1的话可以增大K到$N - \max(dig_i, K) + K$。

但现在dig是不确定的，考虑DP一下。设$F[i][j]$表示从高到低考虑了$\ge i$的位，现在的K是j，最小的对$i-1$位的退位（这个退位是因为高位太大了，我们可能会把某些i位的2个1换成$i-1$位的4个1，这样可以令K更大），易知任意时刻$F[i][j]$都是偶数。转移的话就考虑$[2^{i-1}]M$和基本方案中的dig_{i-1}，注意到我们枚举了K由j变大到j'的话，可以直接确定所需最小的退位值。最后合法当且仅当存在$F[0][i] = 0$。

这个DP有一维是$\mathcal{O}(N)$的，只能通过$N \le 10$的数据，考虑优化它。注意到任意时刻如果已经有了$F[i][j] = 0$（$j \ge 3$），那么可以直接确定合法（因为没有退位，并且后面的基本方案中的$dig_i \le 3$）。还可以发现一个性质，如果有一个转移是$F[i][j]$转移到了$F[i-1][j']$，使得$F[i-1][j'] > 0$且$j' \ge 5, j' - j \ge 2$的话，这个转移是无用的。证明考虑到我们可以转移到$F[i-1][j'-2]$，这里$j'-2 \ge 3$，且若$F[i-1][j'] = 2$有$F[i-1][j'-2] = 0$，直接合法，否则有$F[i-1][j'-2] = F[i-1][j'] - 4$，归纳一下可知不会变劣。

那么我们的DP状态第二维上限就是$\mathcal{O}(\log_2 S)$的了（转移到$j' \ge 5$之后每位至多增加1），并且可以发现每个$F[i][j]$满足上述条件的转移数目是$\mathcal{O}(1)$的，那么只转移有效状态可以做到单次DP$\mathcal{O}(\log^2 S)$的复杂度。

算上二分总时间复杂度为$\mathcal{O}(T \log^3 S)$。

中位数

原题链接

当$P=00010111$的时候就是AGC022 E，那个题目有个自动机的做法，考虑推广到任意的P上。

令串S能变成$'1'$当且仅当$f(S)=1$。我们定义两个串A和B是等价的当且仅当对于任意的S，都有$f(A+S)=f(B+S)$，显然等价关系满足自反性，对称性和传递性，因此把所有的串划分成了等价类。这样性质是很好的：A和B等价可以推出$A+S$和$B+S$等价。因此我们容易用BFS构造出一个DFA，DFA上每个节点是一个等价类。但是问题在于判定两个串A和B等价的时候我们不能测试任意的S，不过我们可以选择测试所有$|S|\leq 10$的01串，如果对于所有这些S都有$f(A+S)=f(B+S)$那么就认为A和B等价。

现在有了这个自动机，我们就可以在上面跑DP了，复杂度是$\mathcal{O}(\sum|S_i|M)$的，这里M是构造出来的自动机点数，实测对所有P有$M\leq 35$。

你可能会问怎么证明这是对的呢？这里有一个验证算法：我们先得到两个$DFA\ A_0$和A_1，它们分别能对长度$\leq L$的01串判定能否变成$'0'$和$'1'$（也可能输入其他的串，但不保证结果正确），那么根据定义，利用A_0和A_1容易得到两个$\epsilon-NFA\ B_0$和B_1，分别能够对长度$\leq L+2$的01串判定能否变成$'0'$和$'1'$（也可能输入其他的串，但不保证结果正确）。如果我们将B_0和B_1转换成对应的DFA，发现它们跟原来的A_0和A_1等价，那么就说明A_0和A_1能够对任意长度的串正确判定了。对于所有的P，让刚刚算法求出来的DFA用这个判定算法判定，都可以证明是对的。