使用雷德算法实现倒位序：

  对于自然顺序（二进制）我们是在低位加 1 得到下一位数，对于倒位序我们是在高位加 1 向低位进位。比如已知一个倒位序数是J求其下一个倒位序数，N位总数 ，把J与N/2比较若J<N/2则J的最高位为 0 ，把最高位置 1 ，就得到了J的下一个倒位序数；若J>=N/2则说明J的最高位为1 ，把最高位置0 ，比较次高位，若次高位为0 ，则把次高位置1，得到J的下一个倒位序，若次高位为1  ， 则把次高位置0,以此类推.

以N = 8 为例：

倒位数顺序                  倒位数                  十进制

    000                           000                          0

    001                           100                          4

    010                            010                         2

    011                            110                         6

    100                            001                         1

    101                            101                         5

    110                            011                         3

111                            111                         7

达到的目的，就是把0到N-1得2进制左右对称的换一下。

这个算法时间估计也是n\*logn，还不如一个个换