算法提高 题目 2 密码锁

时间限制：1.0s   内存限制：1.0GB

问题描述

　　你获得了一个据说是古代玛雅人制作的箱子。你非常想打开箱子看看里面有什么东西,但是不幸的是,正如所有故事里一样,神秘的箱子出现的时候总是会挂着神秘的锁。  
　　这个锁上面看起来有 *N* 个数字,它们排成一排,并且每个数字都在 0 到 2 之间。你发现你可以通过锁上的机关来交换**相邻**两个数字的顺序。比如,如果原来有 5 个数字 02120,在一次交换以后你就可以得到 20120,01220,02210 或者 02102。  
　　根据你所搜集的情报,这个锁在上面存在某连续四个数字是“2012”的时候会自动打开。现在,你需要计算一下,你至少需要进行多少次交换操作才能打开这把锁?

输入格式

　　输入数据的第一行有一个正整数 *N*。(4 ≤ *N* ≤ 13) 输入数据的第二行有 *N* 个数字 *a*1,*a*2, ..., *aN* ,其中 *ai* 表示这个锁上面第 i 个数字的值,满足 0 ≤ *ai* ≤ 2。这些数字之间没有空格分隔。

输出格式

　　你只需要输出一个数字,即你至少需要的交换次数。如果无论如何都没有希望打开这把锁,输出 -1。

样例输入

5  
02120

样例输出

1

对样例的解释

　　把前两个数字交换以后,锁上的数字是 20120,其中存在连续四个数字2, 0, 1, 2,因此锁会打开。

