## 3.11小题解

# **T1 中位数（median.cpp/c/pas）**

本题数据范围n有十万。根据时间复杂度，多是n log n,或是n的算法。这里我给出的是

O（n）的。

50%分数 

将读入数据简化处理，大于k的记为1，小于k的记为-1，等于k的记为0，并且用p记录这个位置。我们用一个数组记录简化后的前缀和。用普遍的区间操作来做。两个for循环，i:1->p,j:p->n。判断区间[i,j]和是否为0是的话ans++；

100%分数 

根据50%分数的做法。我们同样是记录前缀和。不过要开一个桶f[sum][2]数组分别记录在p前与在p后当前缀和为sum的数量。不过由于c++中数组下标不能为负数。只能再在sum的基础上加个n。根据区间和的思想，想让区间[i,j]和为0时，显然前缀i-1和，前缀j和要相等。根据乘法原理，我们可以得到数目为f[sum][0]\*f[sum][1]的情况。

开个for求一下总和就好了。

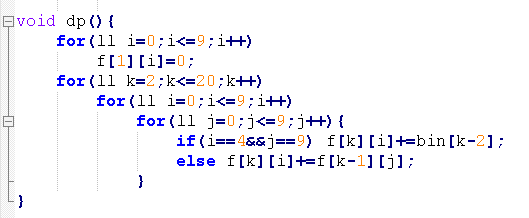
# **T2 幸运数（lucky.cpp/c/pas）**

20%分数 

可以直接从1开始枚举，判断每个数里面是否有49；

100%分数 

显然的数位DP，我们开个f[i][j]数组存i位数j开头的数之前有多少的数含有49。然后就非常的基础。



# **T3 小球（ball.cpp/c/pas）**

这道题非常的水。。。别看数据大，其实暴力就能AC。我们可以简单的证明一下。由于每次倒的时候都需要倒入与原来小球相等的数量，于是最后的一次操作就是两个杯子的小球数量相等。所以他们的小球数量和为偶数。在一开始的时候我们可以这么判断，如果不是则直接-1结束。是的话，如果是两个偶数，怎么倒都不会出现奇数，我们可以先一起除2到两个都是偶数的情况，有一奇一偶同样可以直接-1结束。之后大数减去小数，小数自乘2。操作数+1。继续重复上面的操作，直到有个数为0，输出操作数。

为什么这样的复杂度不会爆炸呢？我们可以用二进制的思想来考虑。当两个数都为奇数时，相当于二进制末尾为1，其中大数减去小数，小数自乘2之后。末尾的两个1都变成了0。对答案没有影响后就可以将这位0去掉，这样每次操作都会少一位，总操作数就不会超过log n，时间复杂度妥妥的。