1. 第K小数(number)

20分做法：

列出所有乘积，排序。

75分做法：

分别将两个数组从小到大排序，二分答案，判断的时候枚举第一个数组中N个数，对每一个选出的数，二分第二个数组。时间复杂度O(NlogQlogM)。

100分做法：

在75分做法中，第二个数组不需要二分，随着第一个数组选出的数不断增大，第二个数具有单调性。时间复杂度O(NlogQ)

另：针对特殊数据的处理，例如40%的数据元素≤10^5，50%的数据K很小，一些数据N相对较小。

1. abcd(abcd)

20分做法：

暴力枚举e数组，判断是否满足条件，若满足，更新最优值。

60分做法：

对条件做变换，0≤num[i]≤b[i]-a[i]，令num[i]=e[i]-a[i]

SUM(e[i]\*c[i])=SUM((num[i]+a[i])\*c[i])=0，

即 SUM(num[i]\*c[i])=SUM(-a[i]\*c[i])

需要求最大值SUM(e[i]\*d[i])=SUM(num[i]\*d[i])+SUM(a[i]\*d[i])

于是发现b[i]-a[i]是物品数量限制，num[i]是i物品的选取数量，c[i]是物品大小，SUM(-a[i]\*c[i])是背包容量，d[i]是物品价值。原问题变成了多重背包问题。若用 N\*物品数量限制\*背包容量的方法可得60分。

100分做法：

采用单调队列优化60分做法，时间复杂度O(N\*背包容量）。

1. 好数(good)

20分做法：

直接暴力判断区间内每个数是否是好数。

40分做法：

可以找规律发现好数的性质：它必须是质数、2的k次幂或者6。证明：如果不为质数，那么一定有2因子，否则1和2都与它互质。设为m\*2^k(m是大于1的奇数),那么m+2和m-2与它互质，1和m\*2^k – 1与它互质，数列确定为1,5,9,13,…，只有这个数为6的时候成立。

于是可以预处理出好数，处理操作的时间复杂度O(N\*M)

70分做法：

发现没有操作2，用基本的线段树做。

100分做法：

加上操作2后，可以发现每个数取余后要么不变化，要么小于原来的二分之一，因此每个数X变化的次数至多为log(X)次，因此可以暴力修改它。在线段树上加上一个记录区间最大值的数，如果当前取余的数S大于这段区间最大值，那么就不用在其上取余。