

A

最大数只有 50，这个是个非常重要的条件。

对于交换 $a[x], a[y]$ ，假设 $x < y$ 的话，其实影响的是 $a[x]$ 与 $a[x + 1] \dots a[y - 1]$ 产生的逆序对，还有 $a[y]$ 与 $a[x + 1] \dots a[y - 1]$ 产生的逆序对，可以用 50 个树状数组存储 1-50 的对应数出现位置。对于 $a[x]$ 与 $a[x + 1] \dots a[y - 1]$ 的，枚举每个大于 $a[x]$ 的数在 $[x + 1, y - 1]$ 这段区间中出现的个数， $a[y]$ 同理，或者可以用二维树状数组查询 $(a[x] \rightarrow a[y], x \rightarrow y)$ 这个平面亦可。

B

```
int number[10010] = {0};
for(int i = 1; i <= n; i++) {
    for(int j = 1; j * j <= a[i]; j++) {
        if(a[i] % j) continue;
        number[j]++;
        if(j * j != a[i]) number[a[i] / j]++;
    }
}
for(int i = 10000; i >= 1; i--) if(number[i] >= k) { ans = i; break; }
```

C

```
solve(int n) {
    if(n == 0) return;
    return (n + 1) / 2 + solve(n / 2) * 2;
}
```

D

最基本的后缀数组了，需要理解后缀数组的定义。将 $A + \$ + B$ 做一次 SA，并求出 height 数组

```
for(int i = 1; i <= tot_len; i++)
    if(sa[i - 1] 和 sa[i]属于不同串的) {
        求出前后 A,B 串对应 height 的最大值，ans += height[i] - 最大值即可
    }
```

或者 SAM 亦可，在普通 SAM 计算每个子串数目的功能基础上改一下，加一个 $c[2]$ ， $c[0]$ 表示 A 的， $c[1]$ 表示 B 的子串数目

```
if(c[0] == 1 && c[1] == 1) ans += node->len - node->f->len
```

E

又一道数位统计题目， $dp[i][j]$ 表示数字长度为 i 的时候，最长前缀为 j 位的和， $num[i][j]$ 表示长度为 i 的时候，最长前缀为 j 位的个数，然后直接推过去吧。。不过其实 $dp[i][j]$ 和 $num[i][j]$ 都是直接算的，然后大家代码都好简短了~~

F

某年 SYSU 校赛题目的加强版，或者上周的 F 题，当时打比赛的时候看错了查询的要求，然后就有了这道题 TAT，贴一下 SYSU 的 BBS 几个月前有人问我不用树套树怎么分治的留

言吧，偷懒一下。

特殊的三维偏序问题，将区间看成 $[x,y,val]$ 的形式，删除操作 $val = -1$ ，否则 $val = 1$ ，这样就将删除和插入操作统一了，可以分治来搞

```
solve(l,r)
{
    solve(l,mid);
    solve(mid + 1,r)
    对[l,mid]排序，按照 x
    对[mid + 1,r]排序，按照 x
    for(int i = mid + 1,j = l; i <= r; i++) {
        while(j <= mid && a[j].x < a[i].x) 如果 a[j]的操作类型不是 3, insert(a[j].y,a[j].val),
        j ++, 用栈记录插入到树状数组内容;
        if(a[i].op == 3) add(ans[a[i].id], query(oo) - query(a[i].y);
    }
    while(栈非空) 清理树状数组
}
```

时间复杂度 $n\log n\log n$