T1

50 分暴力

直接暴力枚举计算 $\gcd$ 即可。

正解

想要使得 $\gcd(x,y)$ 最大，一个简单的想法是让 $x=k$，$y=2k$，因为数的范围要不大于 n，那么 $k\le\lfloor\frac{n}{2}\rfloor$。

那么答案就是 $\lfloor\frac{n}{2}\rfloor$，因为如果要让答案 ans 再大的话 $\lfloor\frac{n}{ans}\rfloor\le 1$，无法找出两个不同的数。

T2

30 分

暴力搜索即可。

60 分

显然变换过程中数一定变小，记忆化搜索即可。

正解

显然如果 n 中存在除了 $2,3,5$ 的其他质因子，必定无解。

考虑三种变换的实质。

操作 1：去掉当前数中的 1 个质因子 2。

操作 2： 将当前数中的质因子 2 增加 1 个，去掉当前数中的 1 个质因子 3。

操作 3： 将当前数中的质因子 2 增加 2 个，去掉当前数中的 1 个质因子 5。

那么做法就出来了，一个质因子 2 贡献为 1，质因子 3 贡献为 2，质因子 5 贡献为 3。

复杂度 $\mathcal{O}(T\log n)$。

T3

30 分

暴力枚举交换的两个位置，然后 $\mathcal{O}(n)$ 计算最小代价，时间复杂度 $\mathcal{O}(n^3)$。

50 分

最小代价的计算显然可以优化，不需要每次重新计算，将交换的两个位置的代价重新计算即可，时间复杂度 $\mathcal{O}(n^2)$。

正解

n 这么大显然是没用的，先用个桶统计一下 (S_i,S_{n-i+1})。

考虑没有那个交换操作怎么做，要成为回文串的话，两个对称位置上的字母 x,y，可以是 x 变成 y 也可以是 y 变成 x，或者两个都变成另外一个字母 z，于是可以在 $\mathcal{O}(26^2)$ 下解决问题。

现在有了一个交换操作，但是并不需要真的去 $\mathcal{O}(n^2)$ 去枚举交换的位置，因为该交换操作最多只涉及到了两对对称位置上的字母，其他的都没有影响，于是直接 $\mathcal{O}(26^4)$ 去枚举涉及到的两对对称位置上的字母是什么就可以了。

T4

20 分

暴力搜索不是 0 的位置的值，然后用组合数计算一下加上 0 的方案数，复杂度是拆分数级别的。

40 分

写一个简单 dp 就行，复杂度 $\mathcal{O}(nr^3)$。

$n = 2$

本质上是要求 $\sum_{i=0}^{r}[l \le (i + i \oplus z) \le r]$。

考虑在二进制下使用数位 dp 解决问题，如果从高位向低位 dp，很难满足两数之和在 $l \sim r$ 之间这个限制。

在此之前先解决一个问题，如何从低位到高位比较大小？（判断一个数是否大于等于一个已知的数 x）

可以用一个 tag 来维护，$tag = 1$ 表示在这个数最低的前 i 位时大于等于 x 最低的的前 i 位。

现在新增一位，如果这个数新增的这一位大于 x 新增的这一位，那么 tag 一定为 1。

如果相等，则 tag 的值不变，即看最低的前 i 位比较结果。

如果 这个数新增的这一位小于 x 新增的这一位，那么 tag 一定为 0。

于是就可以从低位向高位 dp，枚举当前这一位填不填，有没有进位，就可以做到 $\mathcal{O}(\log r)$ 的复杂度。

正解

其实跟 $n = 2$ 差不多，无非是有没有进位变成了进位是多少，每次转移枚举 A 中有多少个数当前二进制位为 1。

时间复杂度 $\mathcal{O}(n^2 \log r)$。