《传送门》解题报告

题目大意

给定两个长为 N 的序列 a, b 和两种操作，要求构造一种操作方案使序列 a 变为序列 b 或者是判断无解。

两种操作分别为：

1. 选一个 i ，把 a_{a_i} 减少 1 ，再把 a_i 增加 1 。

2. 选一个 i ，把 a_{a_i} 增加 1 ，再把 a_i 减少 1 。

所有的加减运算都在模 N 意义下进行。

题解

算法一

正确判断是否有解可以得到 10 分。

注意到每个操作都是给序列上一个数加 1 ，一个数减 1 ，所以整个序列的和在模 N 意义下始终不变。

这意味着如果序列 a 中所有数的和与数列 b 中所有数的和在模 N 意义下不相等，那么该问题无解。

还有一种特殊情况，就是当每个 a_i 都等于 i 时，由于此时任何操作都不能使序列发生改变，所以此时如果序列 a 和 b 不相等，那么问题无解。

可以通过提交发现除此之外的情况该问题都有解。

期望得分：10 分

算法二

对于 $N \le 8$ 的数据，可以用广度优先搜索来寻找解。由于一共只有 $O(N^{N-1})$ 种状态，所以广度优先搜索的复杂度是 $O(N^N)$ 。

期望得分：20 分，结合算法一有 28 分。

算法三

由于本题是按照还原正确的位置数量 cnt 给分，cnt 越大分数越高，所以我们可以使用一些非完美算法来得到一些较优的解。

一种方法是这样的：

每次随机一个 i 直到 $a_i \neq i \land a_i \neq b_i$ ，然后直接对 i 操作把 a_i 调整为 b_i 。一直重复这样的操作直到程序运行够一定的时间。

虽然有些调整会破坏一些已经调整好的位置，但是由于每一次破坏只会使原来的值改变 1 ，所以我们可以猜想，在一定的调整次数后会有较多的位置的值是我们需要的，即有较大的 cnt 。

使用该算法可以在第二个子任务中获得 10 分以上，但后面两个子任务几乎只有判断有解的分数。

期望得分：$15 \sim 25$ 分，结合算法二有 $30 \sim 40$ 分。

算法四

可以在此基础上做进一步的优化。

算法三不能跑出更优的解是因为它常常会陷入一个不存在满足上述条件的 i 且 cnt 较小的局面，有一种改进的方法是在出现没有 i 可选的局面时对序列进行微调，这样可以给序列一个新的调整空间。

还有一个问题就是有些操作会使 cnt 大大减少。我们可以把每次操作都记录下来，在操作序列中取一个最优的前缀。

期望得分：$45 \sim 55$ 分，结合算法二有 $60 \sim 70$ 分。

当然，使用不同的非完美算法，可以根据算法的优劣得到不同的分数。

算法五

我们可以发现一个组合操作：当 $a_i \not\equiv i+1 \pmod N$ 时，按顺序执行 `Dec(i)` 和 `Inc(i)` 后，相当于让 a_{a_i} 增加 1 ，让 $a_{(a_i-1) \bmod N}$ 减少 1 。

于是我们可以找到一个使 $a_i \neq i$ 的 i ，然后把 a_i 的值调整为 $(i-1) \bmod N$ ，反复执行 `Dec(i)` 和 `Inc(i)` 直到 $a_{(i-1) \bmod N} = \left(b_{(i-1) \bmod N} - 1\right) \bmod N$ 。接着执行一次 `Dec(i)` 操作，此时就满足了 $a_{(i-1) \bmod N} = b_{(i-1) \bmod N}$ ，并且此时 $a_i = (i-2) \bmod N$ 。

于是我们又可以用类似的方法把 $a_{(i-2) \bmod N}$ 的值改为 $b_{(i-2) \bmod N}$ ，就这样一直迭代下去直到 $a_i = (i+1) \bmod N$ 。这时不能进行类似的操作了，但这样我们保证了 cnt 至少是 $N-2$ 。

这样我们就可以在第 $2,3$ 两个子任务中得到高分。

为了保证在第 $1,4$ 两个子任务中得到较高的分数，需要结合算法二的暴力。

这样我们可以得到 $70 \sim 80$ 分。

算法六

算法五还差 a_i 和 $a_{(i+1) \bmod N}$ 的问题需要解决。显然根据序列和在模 N 意义下不变的性质，解决了一个就意味着解决了另一个。为了方便，我们选择解决 a_i 的问题。

为了让 a_i 等于 b_i ，我们就需要在使 $a_i = i+1$ 以后执行若干次 `Inc(i)` 。

这些操作会使 $a_{(i+1) \bmod N}, a_{(i+2) \bmod N}, \cdots, a_{(i+b_i-1-i) \bmod N}$ 减少 1 。

所以我们可以在之前调整的时候把

$$a_{(i+2) \bmod N}, a_{(i+3) \bmod N}, \cdots, a_{(i+(b_i-1-i) \bmod N) \bmod N}$$

修改为

$$\left(b_{(i+2) \bmod N}+1\right) \bmod N, \left(b_{(i+3) \bmod N}+1\right) \bmod N, \cdots, \left(b_{(i+b_i-1-i) \bmod N}+1\right) \bmod N,$$

然后再执行那些 `Inc(i)` 的操作就可以把 a 序列变为 b 序列了。

这样我们就可以得到 100 分。调用 `Inc` 和 `Dec` 操作的总次数在 $O\left(N^2\right)$ 级别，达到理论下界。