《Y 君的序列》解题报告

华师大二附中 管晏如

2022.10.24

1 题目大意

Y 君有一个长度为 n 的序列 $\{a_n\}$，初始时 $\forall 1 \le i \le n, a_i = i$。

他可以进行若干次如下的操作，目标是将序列转变为另一个给定的序列 $\{b_n\}$，即 $\forall 1 \le i \le n, a_i = b_i$。**保证 $\{b_n\}$ 是一个 1 到 n 的排列**。

- 选择 i, j 满足 $1 \le i, j \le n, i \ne j, 2|a_i$
- 设 $x = a_i/2$，令 a_j 加上 x，a_i 减去 x

Y 君想知道，他的目标是否可以在**有限**步内达成。如果可以，还请你给出一种步数尽量少的操作方案。

注意你**不需要**最小化操作的步数，只要你正确判断了是否有解、给出的方案合法、且步数不超过一个给定的值 M，就算作答案正确。具体的限制可以看【数据范围】。

你还需要保证在操作的过程中，序列中的所有值位于 $[1, 10^9]$ 内。可以证明，如果有解，则一定可以在本题的限制下得出至少一种方案。

2 数据范围

对于所有数据，保证 $1 \le n \le 10^5, 1.5 \times 10^6 \le M \le 1.5 \times 10^7$。

保证对于 $n > 1000$ 的测试点，$\{b_n\}$ 随机生成。

Subtask 1（17pts）：$1 \le n \le 10, M = 10^7$。

Subtask 2（ 7pts）：$1 \le n \le 150, \forall 1 \le i \le n, b_i \equiv (i-1) \pmod n$。

Subtask 3（14pts）：$1 \le n \le 150$。

Subtask 4（16pts）：$1 \le n \le 1000, M = 1.5 \times 10^7$。

Subtask 5（16pts）：$1 \le n \le 5000, M = 1.5 \times 10^7$。

Subtask 6（30pts）：$1 \le n \le 10^5, M = 1.5 \times 10^7$。

时空限制：1s, 512MB

3 题解

3.1 算法一

对于 $n \le 10$，可以采用朴素暴力（如搜索、BFS），这里不赘述。期望得分：17。

3.2 算法二

我们通过对 n 归纳，来证明一定有解。

当 $n=1$ 时不需要操作。下面假设 $n \geq 2$，并设所有长度为 $n-1$ 的排列都可以操作得到。

方便起见，记 $pos[x]$ 表示当前排列中值 x 所在的位置，$des[x]$ 表示目标排列中值 x 所在的位置（$b[des[x]]=x$）。

我们尝试将 n 放到 $des[n]$ 上，然后对 1 至 $n-1$ 使用归纳。

具备特殊性质的 Subtask2 提示我们考虑将所有 a_i 在模 n 意义下 -1。

事实上，这存在 $n-1$ 步的直接构造。以 $n=7$ 为例（阴影标出的是操作位置）：

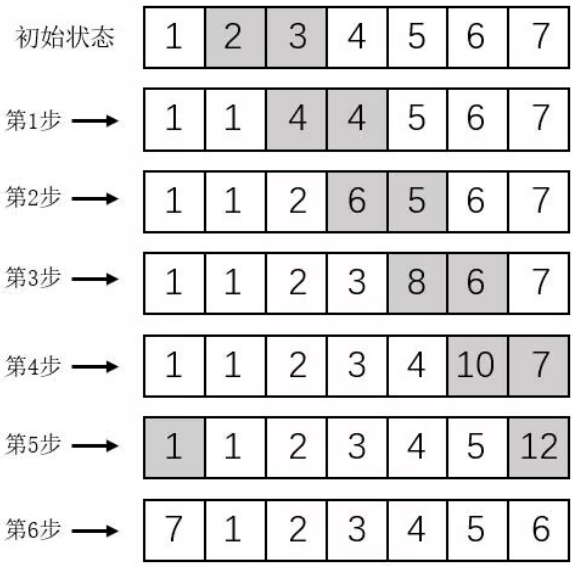

图 1

这样 $n-1$ 步为一轮，则经过 $\leq (n-1)$ 轮，n 会到达位置 $des[n]$。

记操作步数为 $T(n)$，则

$$T(n) \leq \sum_{i=0}^{n-1} i^2 = \frac{1}{6}(n-1)n(2n-1)$$

可以通过 $n \leq 150$ 的部分分，期望得分：38。

3.3 算法三

定义两个数 x 和 y 可以**直接交换**，当且仅当对 x,y 所在位置反复操作，可以在有限步内实现值的交换。

现在的目标仍然是将 n 移动到 $des[n]$ 上，其他的如何移动并不需要关心。

假设我们找到了一个序列 $d_1 = a[des[n]], d_2, ..., d_k = n$，其中 $\forall 1 \le i < k$，d_i 和 d_{i+1} 都可以**直接交换**。那么可以沿着这个顺序，将 $a[des[n]]$ 通过 d_1, d_2 的**直接交换**改成 d_2，通过 d_2, d_3 的**直接交换**改成 $d_3 \cdots\cdots$最后通过 $d_{k-1}, d_k = n$ 的**直接交换**改成 n。

当 $n \le 1000$ 时，我们可以枚举每一对数 x, y，然后暴力操作，判断其是否可以**直接交换**。在实现时可以设定操作步数 $\le 2\lceil \log_2 n \rceil$，或设定中间结果不能超过 $2n$。

接着使用 Floyd 或 Dijkstra 等最短路算法，即可找到以 $a[des[n]]$ 与 n 为端点的路径 $\{d_k\}$。

通过程序验证，可以发现操作步数符合题目要求。这一做法的正确性说明将在**算法五**中看到。

时间复杂度 $\mathcal{O}(n^3)$ 或 $\mathcal{O}(n^2 \log_2 n)$，期望得分：54。

3.4 算法四

我们将针对某两个位置的反复操作进行一定的拓展，给出如下引理：

Lemma 1. *对于正整数 x, y，若 $x \neq y$，$x + y$ 为奇数，则对于初始分别是 x, y 的两个位置不断操作，最终会回到 x, y，即构成纯循环。*

证明 令 $t = x + y$，不妨 x 是偶数，y 是奇数，则第一次操作后 x, y 变为了 $x/2, y + x/2$。

由于

$$y + x/2 \equiv y + (-y)/2 \equiv y/2 \pmod{t}$$

故在模 t 意义下，x 变为 $x/2$ 的同时，y 也变为了 $y/2$。

因此，假设进行了 s 次操作，那么在模 t 意义下，x 变为了 $x/2^s$，y 变为了 $y/2^s$。

又因为 t 是奇数，所以我们可以取 $s = ord_2(t)$，这样 $2^s \equiv 1 \pmod{t}$，经过 s 次操作后一定会回到 x, y。 □

Lemma 1 也告诉我们，在 $x + y$ 是奇数时，x, y 的反复操作具有可逆性。

现在，我们代入 $x = n - 1, y = n$，则 $t = 2n - 1$ 是奇数。注意到 $n - 1, n$ 的逆操作会到达 $2n - 2, 1$，那么我们一定可以对 $n - 1, n$ 顺操作至 $2n - 2, 1$。

随后，操作原序列中未操作过的 1 与现在的 $2n - 2$，就实现了 n 与 1 位置的调换，同时将 $n - 1$ 归位。下面是该过程的示意图（阴影标出的是操作位置）：

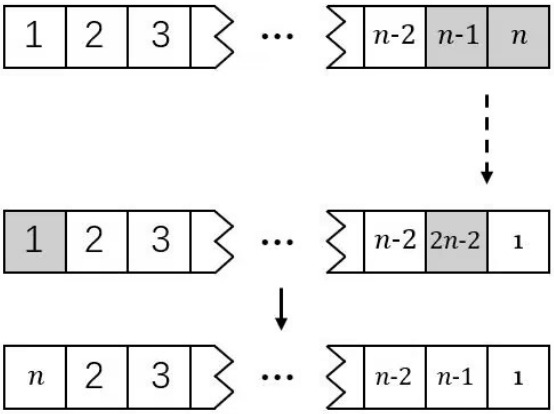

图 2

而将 n 放置到 $des[n]$，只需先交换 1 和 $a[des[n]]$，然后交换 1 和 n。

在给定排列随机的情况下，每对 $i-1$ 与 i 期望产生两次贡献，因此该方法的期望操作步数为 $T(n)=2\sum_{i=1}^{n} ord_2(2i-1)$。

虽然 $T(n)$ 的级别我们不容易直接估计，但是可以考虑编写一段程序来计算：

Function T(n)

```
function ord(n)
    x ← 2
    res ← 1
    while x ≠ 1
        res ← res + 1
        x ← x × 2 % n
    return res
function T(n)
    ans ← 0
    for i from 1 to n
        ans ← ans + ord(i × 2 + 1)
    return ans × 2
```

运行结果显示 $T(5000)$ 约为 1.26×10^7，期望得分：70。

3.5 算法五

现在我们将**算法四**的构造稍加修改，再与**算法**三相结合，最终说明上述最短路的操作步数 $\leq 2\lceil \log_2 n \rceil^2$。

Lemma 2. *设非负整数 s 满足 $y = 2^s + 1 - x > 0$，则对初值分别是 x, y 的两个位置反复操作 s 次，它们会变为 y 和 x，即实现值的交换。*

证明 注意到 $x + y = 2^s + 1$ 是奇数，我们可以沿用 **Lemma 1** 的证明。那么反复操作 s 次后，x 变为：

$$x \cdot 2^{-s} \equiv -x \equiv y \pmod{2^s + 1}$$

这里的本质是 2 模 $2^s + 1$ 的半阶是 s。同理 y 也会变为 x，得证。 □

在 **Lemma 2** 中，代入 $x = n$，并取 s 为满足 $2^s + 1 - n > 0$ 的**最小**非负整数，可知此时 $y = 2^s + 1 - n < n$。我们记这样得到的 y 为 $f(n)$。

因此，每次 n 与 $f(n)$ 交换后，n 会单调减少，直到变成 1。并且每次 s 也是单调减少的，故 $n \to f(n) \to f(f(n)) ... \to 1$ 的过程，至多迭代 $\lceil \log_2 n \rceil$ 次。

算法三中的每一小步，形式化地说，就是把一个 y 放置到当前 x 所在的位置上。那么我们可以先把 1 放置到 x 所在的位置上，然后把 y 放置到 1 所在的位置上。

这两个过程是类似的，前者是 x 不断与 $f(x)$ 交换，即 $x \to f(x) \to ... \to 1$；后者是 1 不断与更大的数交换，直到变成 y，即 $1 \to ... \to f(y) \to y$。

注意这里每个“$\to$”对应 s 次题目中的操作，而 $s \leq \lceil \log_2 n \rceil$，且 s 是单调递减的。

那么，我们一共花费 $\leq s(s+1)/2 \leq (\lceil \log_2 n \rceil + 1)^2$ 次操作，就可以将 n 放置到了 $des[n]$。

从而，$T(n) \leq T(n-1) + (\lceil \log_2 n \rceil + 1)^2$。经过实际随机数据的测试，当 $n = 10^5$ 时，操作次数约为 1.38×10^7，期望得分：100。

注：在整个操作过程中，序列中的值不会超过 2^s，因而也不会超过 $2n$。

4 总结

本题是一道具有一定思维难度的构造题。解题的首要切入点为归纳，随后的构造方法较为多样，且实际需要的操作次数具有一定的梯度。适当使用程序的辅助有利于选手发现和验证更多的性质，不断改进和优化解法。

此外，为了减小输出量和运行常数，本题在实现上采用了交互的方式；最后两个 Subtask 的数据随机是考虑到在不影响数据强度的情况下，确保实际运行结果在常数上小于理论最大值的正确性，同时不作任何针对性地数据构造，以减少选手不必要的顾虑。最终正解方案的构造与合法性并不依赖数据的随机性。

5 参考文献

[1] Multiplicative order, wikipedia,
`https://en.wikipedia.org/wiki/Multiplicative_order`

[2] rosettacode.org - examples of multiplicative order in various languages,
`https://rosettacode.org/wiki/Multiplicative_order#C++`