

构造题选讲 Part1

Scape

March 2, 2019

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

- 我也不知道为什么

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

- 我也不知道为什么
- 杜老师设计的讲课内容会这样地无情

Codeforces 468C Hack it

Introduction

**Codeforces
468C Hack it**

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 令 $f(x)$ 表示整数 x 在十进制表示下各个数位数字之和。

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 令 $f(x)$ 表示整数 x 在十进制表示下各个数位数字之和。
- 给定 a , 请你给出两个整数 l, r ($1 \leq l \leq r < 10^{200}$) 使得 $\sum_{i=l}^r f(i) \equiv 0 \pmod{a}$.

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 令 $f(x)$ 表示整数 x 在十进制表示下各个数位数字之和。
- 给定 a , 请你给出两个整数 l, r ($1 \leq l \leq r < 10^{200}$) 使得 $\sum_{i=l}^r f(i) \equiv 0 \pmod{a}$.
- 保证存在一个解。

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 令 $f(x)$ 表示整数 x 在十进制表示下各个数位数字之和。
- 给定 a , 请你给出两个整数 l, r ($1 \leq l \leq r < 10^{200}$) 使得 $\sum_{i=l}^r f(i) \equiv 0 \pmod{a}$.
- 保证存在一个解。
- $1 \leq a \leq 10^{18}$

Codeforces 468C Hack it

Introduction

**Codeforces
468C Hack it**

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑 x 与 $x + 10^y (x < 10^y)$

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑 x 与 $x + 10^y (x < 10^y)$
- $f(x + 10^y) - f(x) = 1$

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑 x 与 $x + 10^y (x < 10^y)$
- $f(x + 10^y) - f(x) = 1$
- $\sum_{i=x+1}^{10^y+x} f(i) - \sum_{i=1}^{10^y} f(i) = x$

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑 x 与 $x + 10^y (x < 10^y)$
- $f(x + 10^y) - f(x) = 1$
- $\sum_{i=x+1}^{10^y+x} f(i) - \sum_{i=1}^{10^y} f(i) = x$
- 考虑枚举 y , 若 $a - \sum_{i=1}^{10^y} f(i) \bmod a < 10^y$ 则就找到了一个解。

Codeforces 468C Hack it

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑 x 与 $x + 10^y (x < 10^y)$
- $f(x + 10^y) - f(x) = 1$
- $\sum_{i=x+1}^{10^y+x} f(i) - \sum_{i=1}^{10^y} f(i) = x$
- 考虑枚举 y , 若 $a - \sum_{i=1}^{10^y} f(i) \bmod a < 10^y$ 则就找到了一个解。
- 求 $\sum_{i=1}^n f(i)$ 只需要简单地计算每一位上每个数字出现次数。

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack it

**Codeforces
487C Prefix
Product
Sequence**

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑一个序列 a_1, a_2, \dots, a_n , 定义它的前缀积序列是 $a_1 \bmod n, a_1 a_2 \bmod n, \dots, (a_1 a_2 \cdot a_n) \bmod n$ 。

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑一个序列 a_1, a_2, \dots, a_n , 定义它的前缀积序列是 $a_1 \bmod n, a_1 a_2 \bmod n, \dots, (a_1 a_2 \cdot a_n) \bmod n$.
- 现在给定 n , 请你给出一个 n 的排列。

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑一个序列 a_1, a_2, \dots, a_n , 定义它的前缀积序列是 $a_1 \bmod n, a_1 a_2 \bmod n, \dots, (a_1 a_2 \cdots a_n) \bmod n$ 。
- 现在给定 n , 请你给出一个 n 的排列。
- 使得这个排列的前缀积序列是 $[0, 1, 2, \dots, n-1]$ 的一个排列。

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 考虑一个序列 a_1, a_2, \dots, a_n , 定义它的前缀积序列是 $a_1 \bmod n, a_1 a_2 \bmod n, \dots, (a_1 a_2 \cdot a_n) \bmod n$ 。
- 现在给定 n , 请你给出一个 n 的排列。
- 使得这个排列的前缀积序列是 $[0, 1, 2, \dots, n-1]$ 的一个排列。
- $n \leq 10^5$ 。

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack it

**Codeforces
487C Prefix
Product
Sequence**

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- n 为大于 4 的合数则无解。

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- n 为大于 4 的合数则无解。
- 首先 $n! \equiv 0 \pmod{n}$, 因此 $(n-1)! \not\equiv 0 \pmod{n}$

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- n 为大于 4 的合数则无解。
- 首先 $n! \equiv 0 \pmod{n}$, 因此 $(n-1)! \not\equiv 0 \pmod{n}$
- 设 $n = pq$, 若 $p \neq q$, 显然有 $(n-1)! \equiv 0 \pmod{n}$

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- n 为大于 4 的合数则无解。
- 首先 $n! \equiv 0 \pmod{n}$, 因此 $(n-1)! \not\equiv 0 \pmod{n}$
- 设 $n = pq$, 若 $p \neq q$, 显然有 $(n-1)! \equiv 0 \pmod{n}$
- 若 $p = q$, 由于 $n > 4, 2p < n$, 故显然也 gg 了。
- $n = 4$ 的时候给个解?

Codeforces 487C Prefix Product Sequence

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- n 为大于 4 的合数则无解。
- 首先 $n! \equiv 0 \pmod{n}$, 因此 $(n-1)! \not\equiv 0 \pmod{n}$
- 设 $n = pq$, 若 $p \neq q$, 显然有 $(n-1)! \equiv 0 \pmod{n}$
- 若 $p = q$, 由于 $n > 4, 2p < n$, 故显然也 gg 了。
- $n = 4$ 的时候给个解?
- n 为素数的时候, $a_i \equiv \frac{i}{i-1} \pmod{n}$ 。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

**Codeforces
618F Double
Knapsack**

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定两个大小为 n 的可重整数集 A, B ,

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定两个大小为 n 的可重整数集 A, B ,
- 两个数集中的元素均为 $[1, n]$ 中的整数。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定两个大小为 n 的可重整数集 A, B ,
- 两个数集中的元素均为 $[1, n]$ 中的整数。
- 现在要求在两个数集中各找出一个非空子集 (子集也为可重整数集), 满足两个集合中元素的和相等。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

**Codeforces
618F Double
Knapsack**

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 把数集当序列处理。做出两个序列的前缀和 SA_i, SB_i 。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 把数集当序列处理。做出两个序列的前缀和 SA_i, SB_i 。
- 对于每个 $SA_i (0 \leq i \leq n)$ ，我们都找到最大的 j 使得 $SB_j \leq SA_i$ 。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 把数集当序列处理。做出两个序列的前缀和 SA_i, SB_i 。
- 对于每个 $SA_i (0 \leq i \leq n)$ ，我们都找到最大的 j 使得 $SB_j \leq SA_i$ 。
- 这样我们可以得到 $n + 1$ 个 $SA_i - SB_j$ ，而 $SA_i - SB_j$ 的取值只有 n 种，根据鸽巢原理，一定存在两个位置它们的 $SA_i - SB_j$ 相等。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 把数集当序列处理。做出两个序列的前缀和 SA_i, SB_i 。
- 对于每个 $SA_i (0 \leq i \leq n)$ ，我们都找到最大的 j 使得 $SB_j \leq SA_i$ 。
- 这样我们可以得到 $n + 1$ 个 $SA_i - SB_j$ ，而 $SA_i - SB_j$ 的取值只有 n 种，根据鸽巢原理，一定存在两个位置它们的 $SA_i - SB_j$ 相等。
- 而由于 A 中元素不为 0，因此他们对应的 j 的位置一定不同，这样我们就找到了一组合法解。

Codeforces 618F Double Knapsack

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 把数集当序列处理。做出两个序列的前缀和 SA_i, SB_i 。
- 对于每个 $SA_i (0 \leq i \leq n)$ ，我们都找到最大的 j 使得 $SB_j \leq SA_i$ 。
- 这样我们可以得到 $n + 1$ 个 $SA_i - SB_j$ ，而 $SA_i - SB_j$ 的取值只有 n 种，根据鸽巢原理，一定存在两个位置它们的 $SA_i - SB_j$ 相等。
- 而由于 A 中元素不为 0，因此他们对应的 j 的位置一定不同，这样我们就找到了一组合法解。
- 时间复杂度 $O(n)$

ONTAK2014 SUM

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

**ONTAK2014
SUM**

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个长度为 n 的正整数集 A 。

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个长度为 n 的正整数集 A 。
- 要求你给出一个大小不超过 n 的正整数集 B , 满足 A 中每个元素都是 B 的某个子集的和且 B 所有的自己的和互不相同。

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个长度为 n 的正整数集 A 。
- 要求你给出一个大小不超过 n 的正整数集 B , 满足 A 中每个元素都是 B 的某个子集的和且 B 所有的自己的和互不相同。
- B 中元素大小不超过 10^{17}

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个长度为 n 的正整数集 A 。
- 要求你给出一个大小不超过 n 的正整数集 B , 满足 A 中每个元素都是 B 的某个子集的和且 B 所有的自己的和互不相同。
- B 中元素大小不超过 10^{17}
- $n \leq 21, 1 \leq A_i \leq 10^{17}$

ONTAK2014 SUM

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

**ONTAK2014
SUM**

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑一个将问题规约到更小规模的方法。

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑一个将问题规约到更小规模的方法。
- 假设当前集合中存在的最小奇数为 k ，我们将集合中所有其他的奇数减去 k 然后将 k 从集合中删去

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑一个将问题规约到更小规模的方法。
- 假设当前集合中存在的最小奇数为 k ，我们将集合中所有其他的奇数减去 k 然后将 k 从集合中删去
- 剩下的数都是偶数，我们去重后将它们除以 2 递归下去

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑一个将问题规约到更小规模的方法。
- 假设当前集合中存在的最小奇数为 k ，我们将集合中所有其他的奇数减去 k 然后将 k 从集合中删去
- 剩下的数都是偶数，我们去重后将它们除以 2 递归下去
- 当前集合为空时则开始递归回去

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑一个将问题规约到更小规模的方法。
- 假设当前集合中存在的最小奇数为 k ，我们将集合中所有其他的奇数减去 k 然后将 k 从集合中删去
- 剩下的数都是偶数，我们去重后将它们除以 2 递归下去
- 当前集合为空时则开始递归回去
- 递归回来时把现在集合 B 中所有元素乘 2，若这层删掉过奇数 k ，那么现在我们在 B 中加入数 k 。

ONTAK2014 SUM

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑一个将问题规约到更小规模的方法。
- 假设当前集合中存在的最小奇数为 k ，我们将集合中所有其他的奇数减去 k 然后将 k 从集合中删去
- 剩下的数都是偶数，我们去重后将它们除以 2 递归下去
- 当前集合为空时则开始递归回去
- 递归回来时把现在集合 B 中所有元素乘 2，若这层删掉过奇数 k ，那么现在我们在 B 中加入数 k 。
- 我们考虑这个过程为何能满足条件

ONTAK2014 SUM

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

**ONTAK2014
SUM**

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

ONTAK2014 SUM

- 每次我们往 B 中添加的元素，都是原来删去的 A 中的元素， A 中删去的元素最后一定会被添加进 B ，因此它肯定能满足第一个条件

ONTAK2014 SUM

- 每次我们往 B 中添加的元素，都是原来删去的 A 中的元素， A 中删去的元素最后一定会被添加进 B ，因此它肯定能满足第一个条件
- 我们模拟递归回去的过程来证明满足第二个条件

ONTAK2014 SUM

- 每次我们往 B 中添加的元素，都是原来删去的 A 中的元素， A 中删去的元素最后一定会被添加进 B ，因此它肯定能满足第一个条件
- 我们模拟递归回去的过程来证明满足第二个条件
- 首先递归到最后，空集满足条件

ONTAK2014 SUM

- 每次我们往 B 中添加的元素，都是原来删去的 A 中的元素， A 中删去的元素最后一定会被添加进 B ，因此它肯定能满足第一个条件
- 我们模拟递归回去的过程来证明满足第二个条件
- 首先递归到最后，空集满足条件
- 递归回来时集合 B 是满足条件的，那么我们将其所有元素都乘 2，这时依然满足条件

ONTAK2014 SUM

- 每次我们往 B 中添加的元素，都是原来删去的 A 中的元素， A 中删去的元素最后一定会被添加进 B ，因此它肯定能满足第一个条件
- 我们模拟递归回去的过程来证明满足第二个条件
- 首先递归到最后，空集满足条件
- 递归回来时集合 B 是满足条件的，那么我们将其所有元素都乘 2，这时依然满足条件
- 递归回来时把现在集合 B 中所有元素乘 2，若这层删掉过奇数 k ，那么现在我们在 B 中加入数 k 。

ONTAK2014 SUM

- 每次我们往 B 中添加的元素，都是原来删去的 A 中的元素， A 中删去的元素最后一定会被添加进 B ，因此它肯定能满足第一个条件
- 我们模拟递归回去的过程来证明满足第二个条件
- 首先递归到最后，空集满足条件
- 递归回来时集合 B 是满足条件的，那么我们将其所有元素都乘 2，这时依然满足条件
- 递归回来时把现在集合 B 中所有元素乘 2，若这层删掉奇数 k ，那么现在我们在 B 中加入数 k 。
- 我们向 B 中加入新元素时， B 中元素一定都为偶数，且加入的新元素为奇数，那么新产生的子集和为奇数，与原来的子集和一定不相等。原来的子集和互不相等那么新产生的子集和互相之间也不相等。这样我们就满足了第二个条件

Codeforces 578E Walking

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

**Codeforces
578E Walking**

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个长度为 n 的字符串表示足迹序列，每个字符为 'L' 或 'R'，分别表示这个足迹是左脚的还是右脚的，已知人走路总是左右脚交替的，且一步可以向任意方向迈出任意距离。求这个人走出这个足迹序列最少需要回几次头（往回迈）。

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个长度为 n 的字符串表示足迹序列，每个字符为 'L' 或 'R'，分别表示这个足迹是左脚的还是右脚的，已知人走路总是左右脚交替的，且一步可以向任意方向迈出任意距离。求这个人走出这个足迹序列最少需要回几次头 (往回迈)。
- $n \leq 10^5$

Codeforces 578E Walking

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

**Codeforces
578E Walking**

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUM**Codeforces
578E Walking**Codeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 若 L 的数量与 R 的数量差大于 1 则无解

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 若 L 的数量与 R 的数量差大于 1 则无解
- 若我们需要回 k 次头, 则我们可以把序列分为 $k+1$ 个合法子序列

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 若 L 的数量与 R 的数量差大于 1 则无解
- 若我们需要回 k 次头，则我们可以把序列分为 $k + 1$ 个合法子序列
- 反过来，我们若能将序列分成 k 个合法子序列，则我们可以构造一个方案使得我们回头不超过 $k - 1$ 次

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 若 L 的数量与 R 的数量差大于 1 则无解
- 若我们需要回 k 次头, 则我们可以把序列分为 $k+1$ 个合法子序列
- 反过来, 我们若能将序列分成 k 个合法子序列, 则我们可以构造一个方案使得我们回头不超过 $k-1$ 次
- 证明: LL, RR, LR, RL 型的子序列个数分别为 a, b, c, d , 首先 $|a - b| \leq 1$ 。

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 若 L 的数量与 R 的数量差大于 1 则无解
- 若我们需要回 k 次头, 则我们可以把序列分为 $k+1$ 个合法子序列
- 反过来, 我们若能将序列分成 k 个合法子序列, 则我们可以构造一个方案使得我们回头不超过 $k-1$ 次
- 证明: LL, RR, LR, RL 型的子序列个数分别为 a, b, c, d , 首先 $|a - b| \leq 1$ 。
- 我们可以先用不超过 $a + b - 1$ 步将所有 LL, RR 合并为 1 个串, 这时这个串的类型有四种可能

Codeforces 578E Walking

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

**Codeforces
578E Walking**

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们可以用不超过 $c - 1$ 步将所有 LR 合并成一个 LR , 对 RL 也类似, 这样我们现在只剩下 LR 与 RL

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们可以用不超过 $c - 1$ 步将所有 LR 合并成一个 LR , 对 RL 也类似, 这样我们现在只剩下 LR 与 RL
- 考虑将某个序列的最后一个字符接到另一个序列上去, 这样我们就得到了一个 LL 与一个 RR

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们可以用不超过 $c - 1$ 步将所有 LR 合并成一个 LR , 对 RL 也类似, 这样我们现在只剩下 LR 与 RL
- 考虑将某个序列的最后一个字符接到另一个序列上去, 这样我们就得到了一个 LL 与一个 RR
- 不论之前 LL, RR 合并剩下的是什么类型的串, 我们总能使用最多两步将它们与现在的两个串接起来

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们可以用不超过 $c - 1$ 步将所有 LR 合并成一个 LR , 对 RL 也类似, 这样我们现在只剩下 LR 与 RL
- 考虑将某个序列的最后一个字符接到另一个序列上去, 这样我们就得到了一个 LL 与一个 RR
- 不论之前 LL, RR 合并剩下的是什么类型的串, 我们总能使用最多两步将它们与现在的两个串接起来
- 这样我们最多使用 $a + b - 1 + c - 1 + d - 1 + 2 = k - 1$ 步

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们可以用不超过 $c - 1$ 步将所有 LR 合并成一个 LR , 对 RL 也类似, 这样我们现在只剩下 LR 与 RL
- 考虑将某个序列的最后一个字符接到另一个序列上去, 这样我们就得到了一个 LL 与一个 RR
- 不论之前 LL, RR 合并剩下的是什么类型的串, 我们总能使用最多两步将它们与现在的两个串接起来
- 这样我们最多使用 $a + b - 1 + c - 1 + d - 1 + 2 = k - 1$ 步
- 剩下的问题就是如何将原序列分成尽量少的合法序列

Codeforces 578E Walking

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

**Codeforces
578E Walking**

Codeforces
472F Design
Tutorial:
Change the
Goal

Codeforces
612E Square
Root of
Permutation

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUM**Codeforces
578E Walking**Codeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 这实际上是一个最小路径覆盖问题，我们可以拆点匹配

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 这实际上是一个最小路径覆盖问题，我们可以拆点匹配
- 但是我们发现这个匹配实际上可以贪心

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 这实际上是一个最小路径覆盖问题，我们可以拆点匹配
- 但是我们发现这个匹配实际上可以贪心
- 每个点与当前所能匹配点中最前面的匹配即可

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 这实际上是一个最小路径覆盖问题，我们可以拆点匹配
- 但是我们发现这个匹配实际上可以贪心
- 每个点与当前所能匹配点中最前面的匹配即可
- 可以发现任意解进行调整总能满足上面条件

Codeforces 578E Walking

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 这实际上是一个最小路径覆盖问题，我们可以拆点匹配
- 但是我们发现这个匹配实际上可以贪心
- 每个点与当前所能匹配点中最前面的匹配即可
- 可以发现任意解进行调整总能满足上面条件
- 时间复杂度 $O(n)$

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

**Codeforces
472F Design
Tutorial:
Change the
Goal**

Codeforces
612E Square
Root of
Permutation

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E Walking**Codeforces
472F Design
Tutorial:
Change the
Goal**Codeforces
612E Square
Root of
Permutation

- 给定两个长度为 n 的数组。

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定两个长度为 n 的数组。
- 每次可以选定 i, j (可以相等), 令 $x_i = x_i \text{ XOR } x_j$

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定两个长度为 n 的数组。
- 每次可以选定 i, j (可以相等), 令 $x_i = x_i \text{ XOR } x_j$
- 最后使得 $x_i = y_i$ 。 10^6 次操作次数。

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定两个长度为 n 的数组。
- 每次可以选定 i, j (可以相等), 令 $x_i = x_i \text{ XOR } x_j$
- 最后使得 $x_i = y_i$ 。 10^6 次操作次数。
- $1 \leq n \leq 10000, 0 \leq x_i, y_i \leq 10^9$

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

**Codeforces
472F Design
Tutorial:
Change the
Goal**

Codeforces
612E Square
Root of
Permutation

Codeforces 472F Design Tutorial: Change the Goal

- 注意到异或操作两个特性

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E Walking**Codeforces
472F Design
Tutorial:
Change the
Goal**Codeforces
612E Square
Root of
Permutation

Codeforces 472F Design Tutorial: Change the Goal

- 注意到异或操作两个特性
- 1. 存在逆操作就是它本身

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 注意到异或操作两个特性
- 1. 存在逆操作就是它本身
- 2. 它可以交换两个数。
$$a = a \text{ xor } b, b = b \text{ xor } a, a = a \text{ xor } b$$
- 利用特性 2 我们可以将 x_i, y_i 按一定顺序消成一组基，
而利用特性 1 我们可以将消成基的数组还原回去

Codeforces 472F Design Tutorial: Change the Goal

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 注意到异或操作两个特性
- 1. 存在逆操作就是它本身
- 2. 它可以交换两个数。
$$a = a \text{ xor } b, b = b \text{ xor } a, a = a \text{ xor } b$$
- 利用特性 2 我们可以将 x_i, y_i 按一定顺序消成一组基，而利用特性 1 我们可以将消成基的数组还原回去
- 我们将两个数组都消成基得到 a_i, b_i ，随后用若干次操作将 a 转化成 b ，若 b 中有数不能由 a 表示出来则无解。随后将 b 还原回 y 即可

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

**Codeforces
612E Square
Root of
Permutation**

Codeforces 612E Square Root of Permutation

Codeforces 612E Square Root of Permutation

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个 n 的排列 p_i , 请你给出一个 n 的排列 q_i , 使得 $q_{q_i} = p_i (1 \leq i \leq n)$

Codeforces 612E Square Root of Permutation

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 给定一个 n 的排列 p_i , 请你给出一个 n 的排列 q_i , 使得
$$q_{q_i} = p_i (1 \leq i \leq n)$$
- $1 \leq n \leq 1000000$

Introduction

Codeforces
468C Hack it

Codeforces
487C Prefix
Product
Sequence

Codeforces
618F Double
Knapsack

ONTAK2014
SUM

Codeforces
578E Walking

Codeforces
472F Design
Tutorial:
Change the
Goal

**Codeforces
612E Square
Root of
Permutation**

Codeforces 612E Square Root of Permutation

Codeforces 612E Square Root of Permutation

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑对于一个排列 q_i 建一张图，图上的边为 (i, q_i) ，显然这张图会是几个环。

Codeforces 612E Square Root of Permutation

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑对于一个排列 q_i 建一张图，图上的边为 (i, q_i) ，显然这张图会是几个环。
- 我们再来考虑排列 q_{q_i} 的图。对于原来 q_i 图中的奇环它们不变，而原来图中的偶环现在会分裂成大小相等的两个环

Codeforces 612E Square Root of Permutation

Introduction

Codeforces
468C Hack itCodeforces
487C Prefix
Product
SequenceCodeforces
618F Double
KnapsackONTAK2014
SUMCodeforces
578E WalkingCodeforces
472F Design
Tutorial:
Change the
GoalCodeforces
612E Square
Root of
Permutation

- 我们考虑对于一个排列 q_i 建一张图，图上的边为 (i, q_i) ，显然这张图会是几个环。
- 我们再来考虑排列 q_{q_i} 的图。对于原来 q_i 图中的奇环它们不变，而原来图中的偶环现在会分裂成大小相等的两个环
- 因此我们考虑找出排列 p_i 中的所有环，若是奇环则将环“来回”走两步。若是偶环则需要找到一个与它大小相等的环进行合并，找不到就无解。

Codeforces 612E Square Root of Permutation

- 我们考虑对于一个排列 q_i 建一张图，图上的边为 (i, q_i) ，显然这张图会是几个环。
- 我们再来考虑排列 q_{q_i} 的图。对于原来 q_i 图中的奇环它们不变，而原来图中的偶环现在会分裂成大小相等的两个环
- 因此我们考虑找出排列 p_i 中的所有环，若是奇环则将环“来回”走两步。若是偶环则需要找到一个与它大小相等的环进行合并，找不到就无解。
- $O(n)$