1067. Sort with Swap(0,\*) (25)

时间限制

150 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

Given any permutation of the numbers {0, 1, 2,..., N-1}, it is easy to sort them in increasing order. But what if Swap(0, \*) is the ONLY operation that is allowed to use? For example, to sort {4, 0, 2, 1, 3} we may apply the swap operations in the following way:

Swap(0, 1) => {4, 1, 2, 0, 3}  
Swap(0, 3) => {4, 1, 2, 3, 0}  
Swap(0, 4) => {0, 1, 2, 3, 4}

Now you are asked to find the minimum number of swaps need to sort the given permutation of the first N nonnegative integers.

**Input Specification:**

Each input file contains one test case, which gives a positive N (<=105) followed by a permutation sequence of {0, 1, ..., N-1}. All the numbers in a line are separated by a space.

**Output Specification:**

For each case, simply print in a line the minimum number of swaps need to sort the given permutation.

**Sample Input:**

10 3 5 7 2 6 4 9 0 8 1

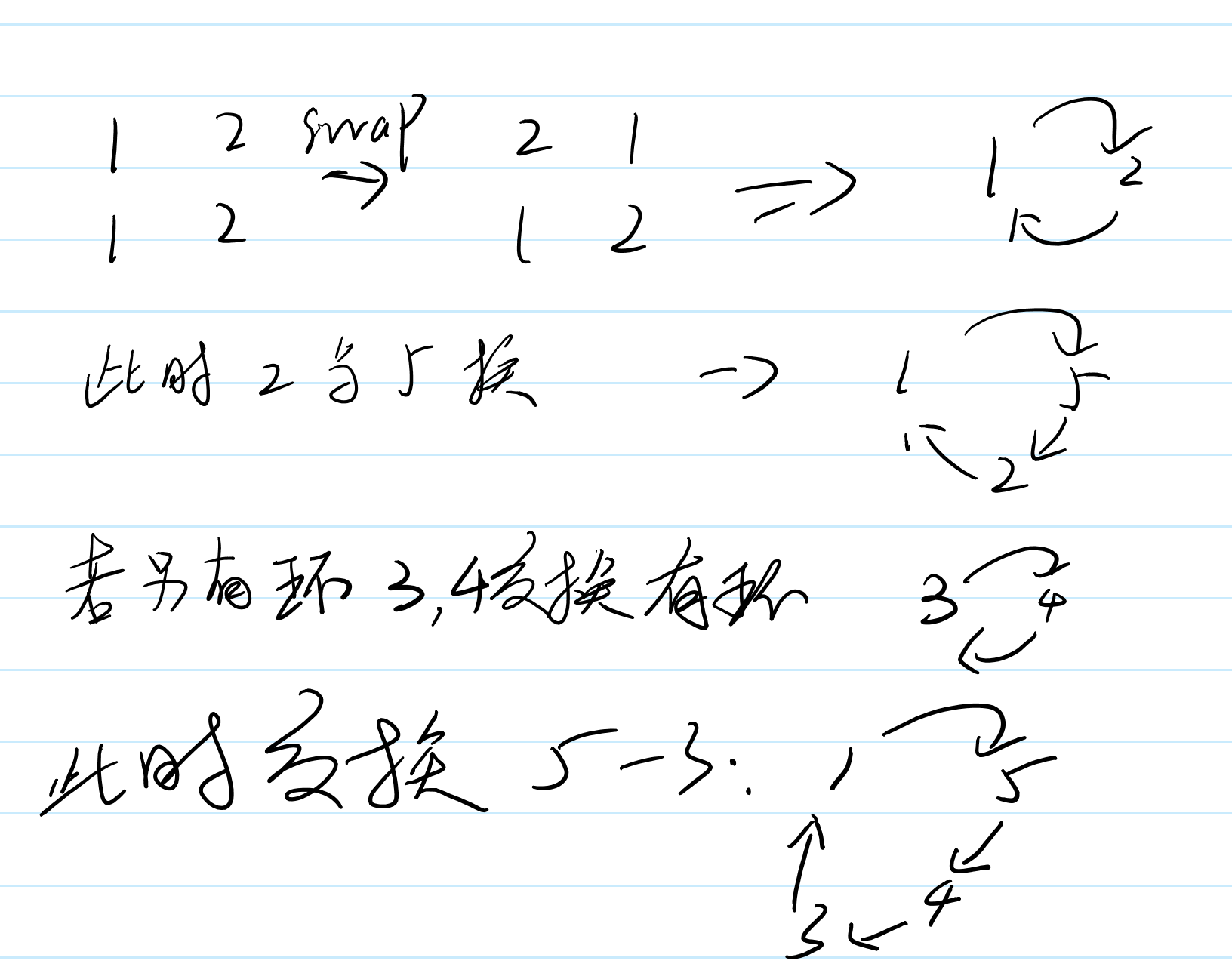
**Sample Output:**

9

这道题需要建立一个数字环的模型

打个比方，在数组中找到0以后，我们能做的事情是找到这个位置本来应该放的数字，然后把找到的数字和0交换，然后找到这个位置应该有的值和0交换，直到0出现在它该出现的位置上。（0必然会出现在0的位置上，关于这点可以从数组一开始的交换开始证明，任何两个数字交换，将两个数字再交换一次就能换回，而如果不在其位的数字再和其他在其位的数字进行交换，那么不在其位的位置的当前所在的位置就会交换过来的数字所替代，所以替代过来的数字指向的位置就是当前的目标位，若和不在其位的数字进行交换，（为了方便，将一个定为x，一个定为y）这也就是说该数字必须在一个环内，则目标环断开，y的目标位指向x原来的目标位，x的目标位指向原来y的目标位，这样就实现了两个环的合并）

上文介绍了环的生成，插入以及合并。



如图。