



① 5 3 1 4
5 3 1 4 . break
↑

② 5 3 1 4
↑

5 1 3 4

↑
5 3 4

③ 1 5 3 4
↑

④ 1 5 3 4
↑

⑤ 从头扫描, 缺2. 返回2

3 \Rightarrow $nums[3-1]=1$
1 \Rightarrow $nums[1-1]=5$

break

$num[2] = 2+1=3$ 不入循环

一换



```
1 package Algorithm;
2
3 public class L41 {
4
5     public int firstMissingPositive(int[] nums) {
6         int n = nums.length;
7
8         for(int i = 0; i < n; i++) {
9             //这个地方是为了把数字放在相应的下标位置, 数字2要放在nums[1]的地方
10            while(nums[i] != i + 1) {
11                //这个是因为不可能进行交换, 所以要跳出去
12                if(nums[i] <= 0 || nums[i] >= n)
13                    break;
14                //这块是防止重复循环
15                if(nums[i] == nums[nums[i] - 1])
16                    break;
17
18                int temp = nums[i];
19                nums[i] = nums[temp - 1];
20                nums[temp - 1] = temp;
21            }
22        }
23        //经过上面的步骤, 那么只要是从1到n的出现过的数字都放在了相应的位置, 所以只需要从开始进行判断
24        for(int i = 0; i < n; i++) {
25            if(nums[i] != i + 1) {
26                return i + 1;
27            }
28        }
29        return n + 1;
30    }
31}
```