287. Find the Duplicate Number

Description | Hints | Submissions | Discuss | Solution

Pick One

Given an array *nums* containing *n* + 1 integers where each integer is between 1 and *n* (inclusive), prove that at least one duplicate number must exist. Assume that there is only one duplicate number, find the duplicate one.

Example 1:

```
Input: [1,3,4,2,2]
Output: 2
```

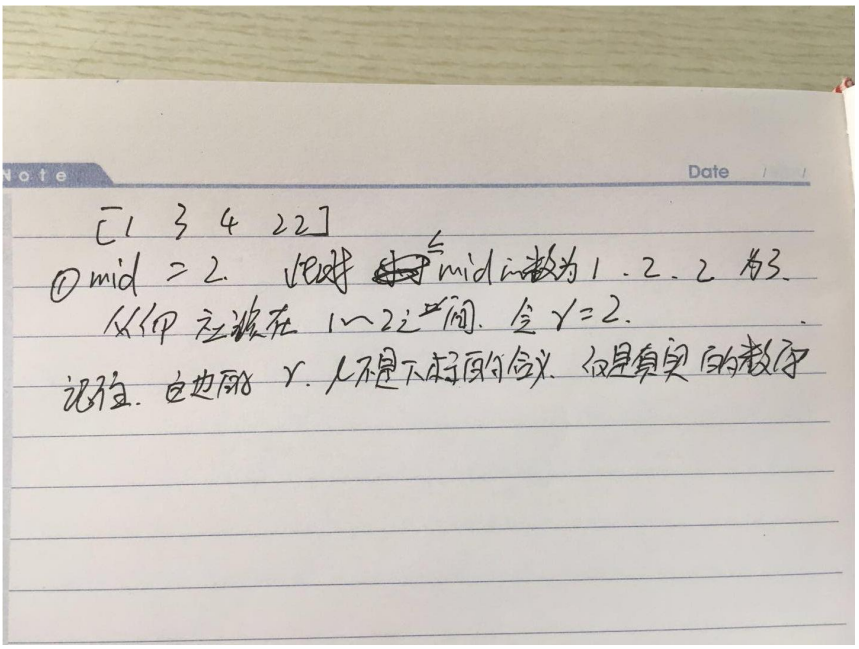
Example 2:

```
Input: [3,1,3,4,2]
Output: 3
```

Note:

1. You **must not** modify the array (assume the array is read only).
2. You must use only constant, $O(1)$ extra space.
3. Your runtime complexity should be less than $O(n^2)$.
4. There is only one duplicate number in the array, but it could be repeated more than once.

Seen this question in a real interview before? Yes No



```java
public class L287 {

      public int findDuplicate(int[] nums) {

    /*
     * 利用抽屉原理，现在数组中求得mid，，如果小于等于mid数目> mid, 那么说明在这个范围内有number 被duplicate了。
     * 所以high = mid - 1. 找到最后duplicate的那个number
     */

       int l=1,r=nums.length-1;
        while(l<r){
            int m=(l+r)/2;
            int c=0;

            for(int i: nums){
                if(i<=m){
                    c++;
                }
            }

            //if c < m,
            if(c>m){
                r=m;
            }else{
                l=m+1;
            }
        }

        return r;}

}
```