Given a sorted array, remove the duplicates [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) such that each element appear only *once* and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

**Example:**

Given *nums* = [1,1,2],

Your function should return length = 2, with the first two elements of *nums* being 1 and 2 respectively.

It doesn't matter what you leave beyond the new length.

题目的意思是找出数组中所有不重复的数字，然后保存在数组的头部。因为java数组固定了大小就无法改变，不可能删除重复元素，只能覆盖掉。

首先我的想法是，i先遍历数组，然后j遍历，找到与nums[i]==nums[j]，然后将后边的所有数据向前移动一位，这样覆盖掉重复的数据，起到了删除的作用。这种想法来源自昨天实现的顺序表的删除。但是值得注意的是，同时你的数组长度也要减去1，因为最后一位是被你舍弃掉的，本也不应该参与循环比较避免出错。

代码：

public int removeDuplicates(int[] nums) {

int counter = nums.length;

for(int i=0;i<counter;i++){

for(int j=0;j<counter&&j!=i;j++){

if(nums[i]==nums[j]){

counter--;

for(int k=j;k<counter;k++){

nums[k]=nums[k+1];

}

}

}

}

for(int i=0;i<counter;i++)

System.out.print(nums[i]+" ");

System.out.println();

return counter;

}

这种方法就比较麻烦，需要两次循环比较，遇到重复的数据的时候同时还需要移动后边的数据向前一位。

答案

答案的思路正好跟我相反，他是去找不重复的数据。首先有i指针和j指针。J指针用于遍历整个数组，i指针则从头开始记录不重复的数据元素。

从头开始，i=0，为数组的第一个元素。J遍历i后边的所有元素，遇到相同的元素略过，遇到不同的数据，i向下走一位，并用该位置记录那个不同的元素nums[i+1]=nums[j]。那么这么会不会覆盖掉不同的元素数值呢？不会的，如果i+1的数据不同于i，那么j也会找到i+1的位置也就是说j=i+1，然后执行nums[i+1]=nums[j]，数据位置不变；如果i+1的数据和i相同呢，j>i+1，就放心覆盖吧，毕竟是重复的数字。j找到了与i不相同的数字保存到i+1的位置，之后j后边的数据跟i元素来说肯定全部都不与nums[i]相同，直接找是否与下一个元素是否相同就行了。然后进行下次循环，令i=i+1，找下一个位置的不相同的数字，j的位置就是刚才那个不重复元素的位置，因为之前的元素肯定与上个元素全部相同，再从头开始找也没有任何意义。直到遍历完一遍整个数组，i之前的数字就就是整个数组无重复的数字。

代码：

public int removeDuplicates\_A(int[] nums){

int i = 0;

for(int j=1;j<nums.length;j++){

if(nums[i]!=nums[j]){

i++;

nums[i]=nums[j];

}

}

for(int j=0;j<i+1;j++)

System.out.print(nums[j]);

System.out.println();

return i+1;

}

得到的启发：就是原本自己想的方法就是记录重复的元素，把重复的元素覆盖掉得到不重复的数组，这样做费时又费力。反方向想就是记录不重复的元素，然后用不重复的元素覆盖重复的元素。省事又省力。

题目给的是sorted array，就是说排好序的数组，答案的方法并不适用于没排好序的数组，因为他安心跳过上个数据nums[i]，找下一个数据nums[i+1]的不同的数据，并不能作用于后边的等于nums[i]的数据。要是没有排序的数组要怎么修改？

在找到与当前元素不同的那个元素的时候，同时需要和i之前的元素进行比较，要是有相同的就跳过，没有相同的才添加。

for(int j=1;j<nums.length;j++){

if(nums[i]!=nums[j]){

int counter=-1; //初始记录i之前没有相同的元素

for(int k=0;k<i+1;k++){ //与i之前的所有数据在进行比较

if(nums[k]==nums[j])

counter=1; //如果有，counter=1

}

if(counter!=1){ //没有的话添加

i++;

nums[i]=nums[j];

}

}

}