Question: <https://leetcode.com/problems/move-zeroes/>

Constraints: ****Note**** that you must do this in-place without making a copy of the array.  
   
We are instructed not to use any extra space but let’s first think what would be happening if we were using extra space.

In such case we would be iterating through the array, and if we found non-zero element we would add that to result array, and thus it would be solved as arrays are initially filled with zeroes.

Now we got an idea that the whole idea is about pointing out the non zero elements and where to place it, so we can use **2-pointer approach**. In that case i) zp - zero pointer ii)i - iterating pointer.

So first we need to set the zp to the first occurring zero’s position.

Then if we find a non zero element then swap the ith position element and zpth position element, and increase the zp pointer by 1.

zp=-1

[0, 1, 0, 3, 12]

For i=0, zp=0

For i=1, zp=0, swap(ar, 0, 1), zp=0+1

[1, 0, 0, 3, 12]

For i=2, zp=1

For i=3, zp=1, swap(ar, 1, 3), zp=1+1

[1, 3, 0, 0, 12]

For i=4, zp=2, swap(ar, 2, 4), zp=2+1

[1, 3, 12, 0, 0]

Solution:  
class Solution {

public void swap(int[] ar, int p1, int p2){

int temp=0;

ar[p1]=ar[p2];

ar[p2]=temp;

return;

}

public void moveZeroes(int[] nums) {

int zp=-1;

for(int i=0; i<nums.length; i++){

if(zp==-1&&nums[i]==0){

zp=i;

}

if(zp!=-1 && nums[i]!=0){

swap(nums, zp, i);

zp++;

}

}

}

}  
Github Link :<https://lnkd.in/ecwtJeaz>