75. sort colors:

I/P: nums = [2, 0, 2, 1, 1, 0]

O/P: [0, 0, 1, 1, 2, 2]

you are given an array with numbers:

0 → red

1 → white

2 → blue

Task:

rearrange the array in-place so that all:

0's come first

then 1's

then 2's

not allowed to use .sort() function.

```
class Solution {
  public void sortColors (int[] nums) {
    int low = 0, mid = 0, high = nums.length - 1;

    while (mid <= high) {
      if (nums[mid] == 0) {
        int temp = nums[low];
        nums[low] = nums[mid];
        nums[mid] = temp;
        low ++;
        mid ++;
      } else if (nums[mid] == 1) {
        mid ++;
      } else {
            int temp = nums[mid];
            nums[mid] = nums[high];
            nums[high] = temp;
             high--;
          }
        }
      }
    }
```

Time → O(n)   Space → O(1).