**Arrays :: Leetcode**

**Easy**

1. how-many-numbers-are-smaller-than-the-current-number

Brute Force:

class Solution {

public int[] smallerNumbersThanCurrent(int[] nums) {

int count=0;

int[] ans=new int[nums.length];

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

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

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

count++;

}

}

ans[i]=count;

count=0;

}

return ans;

}

}

Leetcode Analysis - Runtime: 20 ms, faster than 5.03% of Java online submissions for How Many Numbers Are Smaller Than the Current Number.

Memory Usage: 41.7 MB, less than 100.00% of Java online submissions for How Many Numbers Are Smaller Than the Current Number.

1. find-numbers-with-even-number-of-digits

Brute Force:

class Solution {

public int findNumbers(int[] nums) {

int answer=0;

int check =0;

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

check=containEven(nums[i]);

if(check%2==0){

answer++;

}

}

return answer;

}

public int containEven(int num){

int val=1;

int a=num;

while(a/10!=0){

a=a/10;

val++;

}

return val;

}

}

Leetcode Analysis – Runtime: 2 ms, faster than 15.19% of Java online submissions for Find Numbers with Even Number of Digits.

Memory Usage: 40.9 MB, less than 100.00% of Java online submissions for Find Numbers with Even Number of Digits.

1. create-target-array-in-the-given-order

Brute Force –

class Solution {

public int[] createTargetArray(int[] nums, int[] index) {

ArrayList<Integer> target= new ArrayList<Integer>(index.length);

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

target.add(index[i],nums[i]);

}

int[] ans=new int[target.size()];

for(int j=0;j<target.size();j++){

ans[j]=target.get(j);

}

return ans;

}

}

Leetcode Analysis - Runtime: 2 ms, faster than 25.32% of Java online submissions for Create Target Array in the Given Order.

Memory Usage: 39.7 MB, less than 100.00% of Java online submissions for Create Target Array in the Given Order.

1. missing-number
2. find-common-characters