Sort 0s, 1s and 2s

Difficulty: **Medium** Accuracy: **50.58%** Submissions: **805K+** Points: **4**

Average Time: **10m**

Given an array **arr[]** containing only **0s, 1s, and 2s.** Sort the array in ascending order.

Note: You need to solve this problem without utilizing the built-in sort function.

Examples:

Input: arr[] = [0, 1, 2, 0, 1, 2]
Output: [0, 0, 1, 1, 2, 2]
Explanation: 0s, 1s and 2s are segregated into ascending order.

Input: arr[] = [0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1]
Output: [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2]
Explanation: 0s, 1s and 2s are segregated into ascending order.

Follow up: Could you come up with a one-pass algorithm using only constant

C++ (12) Start Timer

```cpp
class Solution {
  public:
    void sort012(vector<int>& arr) {
        // code here
        int n=arr.size();
        int z=0,i=0,t=n-1;

        while(i<=t)
        {
            if(arr[i]==0)
            {
                swap(arr[z],arr[i]);
                z++;
                i++;
            }
            else if(arr[i]==1){i++;}
            else
            {
                swap(arr[i],arr[t]);
                t--;
            }
        }

    }
};
```

Custom Input Compile & Run

Find H-Index

Difficulty: **Medium** Accuracy: **53.4%** Submissions: **54K+** Points: **4**

You are given an array **citations[],** where each element **citations[i]** represents the number of **citations** received by the **i^{th}** paper of a researcher. You have to calculate the researcher's **H-index**.

The **H-index** is defined as the maximum value **H**, such that the researcher has published at least **H papers**, and all those papers have **citation value** greater than or equal to **H**.

Examples:

Input: citations[] = [3, 0, 5, 3, 0]
Output: 3
Explanation: There are at least 3 papers with citation counts of 3, 5, and 3, all having citations greater than or equal to 3.

Input: citations[] = [5, 1, 2, 4, 1]
Output: 2
Explanation: There are 3 papers (with citation counts of 5, 2, and

C++ (12) Start Timer

```cpp
class Solution {
  public:
    int hIndex(vector<int>& citations) {
        // code here
        int n=citations.size();
        int ans=0;
        sort(citations.begin(),citations.end(),greater<int>()

        for(int i=0;i<n;i++)
        {
            if(citations[i]>=i+1)
            {
                ans++;
            }
        }

        return ans;

    }
};
```

Custom Input Compile & Run Sub