Day-11

11-09-2021

Level : Medium

Q1 Largest Number (list of non-negative integers nums, arrange them such that they form the largest number.

Input: nums = [10, 2]
Output: "210"

logic ⇒ first convert array into string and sort the string by STL library sorting.

//fun create.

```
static bool compare(string &s1, string &s2) {
    return (s1+s2) >= (s2+s1);
}
```

//fun given.

```
string largestNumber(vector<int> &nums) {
int x=0;
for(int i=0; i<nums.size(); i++)
{
    if(nums[i]==0) { x++; }
}
  if(x==nums.size())
  {
      return "0";
  }
```

```
vector<string> s(nums.size(), "");
for(int i=0; i<nums.size(); i++)
{
    s[i] = to_string(nums[i]);
}
sort(s.begin(), s.end(), compare);
string ans = "";
for(int i=0; i<s.size(); i++) {
    ans += (s[i]);
}
return ans;
}
```

Level-hard

Q1 Merge of Two Sorted Arrays. (Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of two sorted arrays. O(log(m+n)).

Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000

```
double findMedian(vector<int>& nums1, vector<int>& nums2)
{
    int n = nums1.size();
    int m = nums2.size();

    vector<double> nums3;
```

```
for (int i=0; i<n; i++) {
    nums3.push_back(nums1[i]);
}
for (int i=0; i<m; i++) {
    nums3.push_back(nums2[i]);
}

sort(nums3.begin(), nums3.end());
int k = nums3.size();
double res = 0;

if (k%2==0); {
    int idx = k/2;
    res = (nums3[idx] + nums3[idx-1])/2;
}
else {
    int idx = k/2;
    res = nums3[idx];
}
return res;
}
```

Leet-medium

Q.2 H-Index (n-h > h)

↓ papers ↓ h citation.

index h have n papers has atleast h citations.

Input: citations = [3,0,6,1,5]
output: 3

Time comp (O(nlogn + n)) = O(nlogn)
↳ neglect

```
int hIndex(vector<int>& citations)
{

    // sort the citations (0,1,3,5,6)
    sort(citations.begin(), citations.end());
    // store the citations in n
    int n = citations.size();
    // Create int i -> index.
    int i;

    // iterate the loop
    for(i=1; i<=n; i++)

    if(citations[n-i] < i)
        break;

    return i-1;
}
```

count: 6 5 3 1 0
paper idx: (1) 2 3 4 5

for 1 count atleast 1 so it is valid index

Same check 2 also valid if equal like for 3 so stop.