# CSC 323 Algorithm Design and Analysis

Spring 2016

Instructor: Dr. Natarajan Meghanathan

## Project 3: Computing h-index using Sorting and Max-Min Formulation

Due: March 3, 2016: 1 PM

The h-index is considered as a measure of both the productivity of a researcher as well as the impact of his/her publications in the research community. A researcher is said to have an h-index of 'h' if s/he has published at least 'h' papers each of which have received at least 'h' citations.

Given an array A of size 'n' (let the number of papers published by a researcher be 'n') whose entries indicate the number of citations received by each paper, your task is to find the h-index of the researcher.

**Steps suggested for implementation**

**Step 1:** Generate an array of *n* integers (ranging from 1 to 100) using a random number generator (see the template in Project 1 for how to generate and use random number generator in Java).

**Step 2:** Sort the input array A using any sorting algorithm of your choice. You need to sort the array in the reverse order (i.e., non-increasing order). You could use a modified version of the insertion sort or merge sort algorithms that we went over in Module 2 to sort an array in reverse order. You could also use a modified version of the simple bubble sort algorithm to sort an array in reverse order (pseudo code given below). Let the sorted array be denoted $A_S$.

Pseudo code of Bubble sort algorithm to sort an array in reverse order

```
for (int j = 0; j ≤ n-2; j++)
    for (int k = 0; k ≤ n - j - 2; k++)
        if (A[k] < A[k+1] )
            Swap(A[k], A[k+1])
```

**Step 3:** Run the sorted array $A_S$ through the following Max-Min formulation test and find the index *i* (*i* runs from 1 to '*n*') that satisfies the formulation: $\underset{i}{Max}\{Min(A_S[i], i)\}$

**Example to Illustrate the Computation of the h-index**

Let an array of 10 integers be: A = {10, 4, 5, 21, 8, 9, 15, 17, 7, 2}

The reverse sorted array $A_S$ = {21, 17, 15, 10, 9, 8, 7, 5, 4, 2}

| Index, *i* | $A_S[i]$ | Min($A_S[i]$, *i*) | Max |
|---|---|---|---|
| 1 | 21 | 1 | 1 |
| 2 | 17 | 2 | 2 |
| 3 | 15 | 3 | 3 |
| 4 | 10 | 4 | 4 |
| 5 | 9 | 5 | 5 |
| 6 | 8 | 6 | 6 |
| 7 | 7 | 7 | 7 |
| 8 | 5 | 5 | 7 |
| 9 | 4 | 4 | 7 |
| 10 | 2 | 2 | 7 |

**h-index = 7**

**Submission**

(1) Submit a hardcopy of your code for generating the array of random integers, sorting and identifying the h-index through the Max-Min formulation steps. Also include a screenshot of the results obtained for an array of 15 random integers.

(2) Submit a desktop-recorded video of your explanation of the code.

**Demo and Reporting:** For demo and reporting purposes, each of you should generate an array of 15 integers (ranging from 1 to 100) using a random number generator (see the template in Project 1 for how to generate and use random number generator in Java).