30 January 2017 -- Computer Architectures -- part 1/2

Matr, Last Name, First Name ....

In the scientific community, the impact of the research of a scientist is often measured by looking and "counting" the number of citations that the scientific papers published by the researcher have received. Currently the two most widely considered metrics are offered by two different providers and are known as Web of Science (WoS) and Scopus (Sco). Due to the different databases used by WoS and Sco, sometimes the results can differ, like in the example here below.

Amir Momeni; Jie Han; Paolo Montuschi; Fabrizio Lombardi (2015)
Design and Analysis of Approximate Compressors for Multiplication. In: IEEE TRANSACTIONS ON COMPUTERS, vol. 64 n. 4, pp. 984-994. - ISSN 0018-9340
Web of Science: 6 - **Scopus:** 11

From a practical point of view, the number of citations is a metrics to evaluate the impact of a single publication. The scientific communities are used to refer to another metrics to evaluate the impact of the overall scientific production of a researcher: the h-index. According to Wikipedia: *The h-index is based on the set of the scientist's most cited papers and the number of citations that they have received in other publications. The definition of the h-index is that* ***a scholar with an h-index of Z has published Z papers such that each one has been cited in other papers at least Z times and at the same time he/she has NOT published Z+1 papers each one cited at least Z+1 times.*** Let us consider the following example of 3 authors each with 6 papers, where citations are the numbers into the boxes).

Author	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5	Paper 6	h-index
John	2	2	2	2	1	5	2
Mary	4	4	5	6	1	0	4
James	1	1	1	0	2	1	1

John has h-index = 2 because it has at least two papers with 2 or more citations, but he does not have 3 papers with 3 or more citations. Mary has h-index=4 because she has papers 1, 2, 3, 4 with at least 4 citations, but not five papers with at least 5 citations (she has only papers 3 and 4 with at least 5 citations). James has only one paper with 2 citations and therefore cannot have h-index=2, but has h-index=1 because he has at least one paper with 1 citation.

Let us assume to have an array of 51 elements CITATIONS DW 51 (?) reporting for one same author the number of citation of her/his most relevant up to 50 papers from years 2001 to 2016, in the following format:

- First cell of the array contains the number **T** of papers (from 1 to 50, i.e. in the next T cells; all other cells are void)
- Second to next **T** cells (at most 50 cells), two bytes encoded as 4+6+6 bits, i.e. yyyy wwwwww ssssss, where yyyy is the year of the paper, offset 2001 (e.g. year 2004 is encoded as 2004-2001=3 → 0011), wwwwww is the binary number of WoS counted citations (from zero to 63) and ssssss is the binary number of Sco citations (0..63)

For the paper in the picture above, the coding is:
Year = 2015 (i.e. 14 offset 2001), WoS citations = 6, Sco citations = 11 → 1110 000110 001011
The array is sorted by year in increasing order (from oldest to most recent year); there is no secondary sub-sorting for WoS or Sco citation numbers.

Tasks to be implemented and corresponding points (only fully completed items will be considered to award points); only one among Items A, B, C, D, E. The additional or bonus items receive **full points** as marked **only** in case the student solved either B, or C, or D or E. Students who solved A, will receive ½ of the points for these items.

- Item A: compute the total numbers separately of **WoS** and **Sco** citations, as well as their average number per paper (i.e., total number of **WoS** or **Sco** citations divided by **T**, without considering fract bits); POINTS → 19
- Item B: compute the h-index of the author by considering only **Sco** numbers; POINTS → 22
- Item C: compute the h-index by considering only **WoS** numbers ; POINTS → 23
- Item D: compute the h-indexes by considering first **WoS** and then **Sco** numbers ; POINTS → 25;
- Item E: compute the h-index both by considering for each paper **max(WoS, Sco)**; POINTS → 30;
- Additional Item 1: receiving in input one year, say **F**, compute the average number of **Sco** citations for all papers published in year **F** (i.e., total number of **Sco** citations obtained by adding up **Sco** citations of all papers published in year **F** divided by the number of papers published in year **F**, without fract bits); POINTS → +3;
- Additional Item 2: receiving in input two years, say **F** and **G**, compute the total number of **Sco** citations for all papers published between year **F** and **G**, bounds included (**F** can be equal to **G**); POINTS → +3;
- Additional Item 3: identify the year where the author has received **overall** the largest number of **Sco** citations, i.e. count all **Sco** citations per year, and then identify the year with more citations; POINTS → +3.
- **Bonus Item**: compute the value obtained by adding up, for all papers, **max(WoS, Sco)**; POINTS → +3.

30 January 2017 -- Computer Architectures –- part 1/2

Matr, Last Name, First Name ....

Please consider that a maximum of 33 points can be accounted here; larger values will be "cut off" to 33.

Depending on the Item(s) solved, the program should receive in input the values of the parameters and should display the requested results; the array CITATIONS is defined and initialized inside the program.

HINTS (observe that)

- **For Students solving Items except A, it is highly recommended to first design and draft the algorithm to compute the h-index. Similarities can be observed among Items B, C and D, while for E the definition of an auxiliary database is advised.**
- Please observe that the maximum number of citations for a paper is 63, and that the maximum number of papers is 50. This information is important to determine what is the required number of bits to host the total number of citations (the choice is left to the Student).
- When computing the average number of citations per paper, Students are NOT requested to compute the fractional part but the integer part only.
- It is advised to design the program as a collection of modules, each one implementing the different Items.

REQUIREMENTS (SHARP)

- It is not required to provide the optimal (shortest, most efficient, fastest) solution, but a working and clear one.
- It is required to write at class time a short and clear explanation of the algorithm used.
- It is required to write at class time significant comments to the instructions.
- Input-output is not necessary in class-developed solution, but its implementation is mandatory for the oral exam.
- Minimum score to "pass" this part is 15 (to be averaged with second part and to yield a value at least 18)

REQUIREMENTS ON THE I/O PART TO BE DONE AT HOME

- The databases (if any) have to be defined and initialized inside the code
- All inputs and outputs should be in readable ASCII form (no binary is permitted).

Please use carbon copy ONLY (NO PICTURES ARE ALLOWED) and retain one copy for home implementation and debug. At the end of the exam please give to professors all the sheets of your solution. Missing or late sheet will not be evaluated. Please provide your classroom submitted solution with several explanatory and significant comments. Please remember that only what has been developed at class time can and will be evaluated at oral time and that it is necessary to write the instructions of the program and not just the description of the algorithm. When coming to oral discussion, please clearly mark in red on your "classroom" copy, all modifications. Please also provide an error-free and running release of the solution, as well as with its printed list of instructions. Please consider that the above are necessary but not sufficient requirements to success the exam, since the final evaluation will be based on a number of parameters. FAILURE TO ACCOMPLISH ALL THE ABOVE NECESSARY REQUIREMENTS WILL CAUSE NO-QUESTION-ASKED AND IMMEDIATE REJECTION.