CSC 323 Algorithm Design and Analysis, Spring 2018
Instructor: Dr. Natarajan Meghanathan
Quiz 5
Implementation of the Greedy Algorithm to Determine an Optimal Allocation of Files in a Tape to Minimize the Average Cost to Access the Files

Due: March 27th, 2018: 1 PM (Submit through Canvas)

In this quiz, you will implement the greedy algorithm discussed in class to determine an optimal allocation of files in a tape to minimize the average cost to access the files.

You would generate an array of 'N' files such that the maximum size of any file is 'M' and the maximum frequency of access for any file is 'F'. You will get the inputs for these three values from the user. The actual size for any file would be a randomly generated integer in the range [1...M] and the actual frequency of access for any file is also a randomly generated integer in the range [1...F].

You need to order the files in the increasing order of (i) File Index, (ii) File Size and (iii) Size / Frequency ratio and determine the average costs to access any file for each of these strategies (as is done in the examples in the slides).

You need to use an appropriate sorting algorithm that would facilitate sorting the dataset based on one of the three measures (file index, file size or size/frequency ratio) at any time. Break any tie in the ordering of the files (when ordered based on file size or size/frequency ratio) using the file index.

Input Values to Use

Student Name	**N**	**M**	**F**		**Student Name**	**N**	**M**	**F**
Leon Anderson	10	25	50		Yoseph H.	10	25	75
Ujjwal Baskota	15	25	50		Antonie Hobson	15	25	75
Albert Boateng	20	25	50		Portia Junius	20	25	75
Nissi Campbell	25	25	50		Justin McGuffee	25	25	75
Samuel A. Dagne	30	25	50		Ryun Moore	30	25	75
James Daniel	10	50	25		Keara Rogers	10	75	25
Zakeia Davis	15	50	25		Timothy Stewart	15	75	25
Justin Epps	20	50	25		Nebiyou Tadesse	20	75	25
Amanuel E. Gebre	25	50	25		Phat Tran	25	75	25
Mel Groom	30	50	25					

Your code should output the following:
(a) The average costs to access the files when ordered based on (i) File Index, (ii) File Size and (iii) Size/Frequency ratio.
(b) The ordering of the files (i.e., print the file index values) based on the allocation strategy (among the three strategies listed above) that results in the lowest average cost to access the files.

Submission:
(1) Your complete code (including the code for the sorting algorithm) as a .java or a .cpp file (python or C# is fine too).
(2) A single screenshot of the outputs (a) and (b), as mentioned above, for the N, M, F values assigned to you.