

CSC 323 Algorithm Design and Analysis, Spring 2019
Instructor: Dr. Natarajan Meghanathan
Quiz 4

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 21st, 2019: 11.59 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
Brown, Demetrius	10	25	50			10	25	75
Cato, Jahelle	15	25	50			15	25	75
Chukwuma, Nzefili	20	25	50			20	25	75
Clark, Armon	25	25	50			25	25	75
Collins, Taylor	30	25	50			30	25	75
Harmon, Alfred	10	50	25		Washington, Daren	10	75	25
Jackson, Martice	15	50	25		Wynn, Marcus	15	75	25
Langat, Vincent	20	50	25			20	75	25
Stewart, Jessica	25	50	25			25	75	25
Tchakoua, Astride	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.