

CS608 - Spring 2018 - Assignment #4

Assigned: March 21st, 2018

Due: April 4th, 2018

The goal of assignments is to give you practice in mastering the course material. Consequently, you are encouraged to collaborate with others. In fact, students who form study groups generally do better on exams than do students who work alone. If you do work in a study group, however, you owe it to yourself and your group to be prepared for your study group meeting. Specifically, you should spend at least 30–45 minutes trying to solve each problem beforehand. If your group is unable to solve a problem, it is your responsibility to get help from the instructor before the assignment is due. **You must write up each problem solution and/or code any programming assignment by yourself** without assistance, even if you collaborate with others to solve the problem. **You are asked to identify your collaborators.** If you did not work with anyone, you must write “Collaborators: none.” If you obtain a solution through research (e.g., on the web), acknowledge your source, but write up the solution in your own words. **It is a violation of this policy to submit a problem solution that you cannot orally explain to the instructor.** No other student may use your solutions; this includes your writing, code, tests, documentation, etc. It is a violation of this policy to permit anyone other than the instructor and yourself read-access to the location where you keep your solutions. **For this assignment, team submissions are allowed. Each team can be composed by at most three members. Only one submission per team is required.**

Submission Guidelines: You have to submit your work on Blackboard by the due date. The assignment has two parts: a set of exercises that are not for grade **but still required**, and a programming assignment that includes a coding part and some non-coding questions. Your solutions for the not-for-grade exercises may be submitted separately in any format (even handwriting). The file with your code must include the header template provided in Blackboard. This header must contain, your name, course number, semester, assignment number, problem number, a list of collaborators (if none, put “none”), and the team members if you submit as a team. Your answers to non-coding questions must be included in this header as well, in the Remarks section.

Then, to complete your submission, you have to upload two files to Blackboard by the due date: a Java source file and a Java class file.

The submission will not be accepted in any other format.

Style and Correctness: Keep in mind that your goal is to communicate. Full credit will be given only to the correct solution which is described **clearly**. Convoluting and obtuse descriptions might receive low marks, even when they are correct. Also, aim for concise solutions, as it will save you time spent on write-ups, and also help you conceptualize the key idea of the problem.

Assignment 4

Programming Assignment Grading Rubric:

The following rubric applies only to the programming assignment.

Program characteristic	Program feature	Credit possible		
Design 30%	Algorithm	30%		
Functionality 30%	Program runs without errors	20%		
	Correct result given	10%		
Input 15%	User friendly, typos, spacing	10%		
	Values read in correctly	5%		
Output 15%	Output provided	10%		
	Proper spelling, spacing, user friendly	5%		
Format 10%	Documentation: name, collaborators, header, etc.	5%		
	Clarity: comments, indentation, etc.	5%		
	TOTAL	100%		

1(20)	2(30)	3(30)	4(20)	TOTAL(100)

Assignment:

Quick Sort is a popular sorting algorithm because it usually achieves optimal $O(n \log n)$ running time in practice. However, in the worst case, the running time can be quadratic. The method to choose the pivot is crucial to avoid the worst cases. The purpose of this homework is to evaluate experimentally the performance of Quick Sort for different methods of choosing the pivot, and compare with Selection Sort, which is simpler but quadratic.

1. **(20 points)** Write a method that implements Selection Sort.
2. **(30 points)** Write methods for two versions of Quick Sort, depending on how the pivot is chosen from each subarray to be sorted, as follows.
 - (a) Choose the pivot from the first position of the subarray.
 - (b) Choose three positions of the subarray at random and make the median of the values in these positions the pivot.
3. **(30 points)** Write a test program that measures the running time of the above three methods while sorting 1000000 numbers (adjust the quantity if needed). Use three types of inputs: already sorted in increasing order, already sorted in decreasing order, and an input with the numbers generated at random. Fill in the following chart with the running times observed.

version	increasing order	decreasing order	random
1			
2a			
2b			

4. **(20 points)** Draw conclusions from the values observed. Are the measurements significantly different for some cases? Why? How the pivot choice helps for those cases where the running time is smaller?

Extra Credit (up to 20%): Obtain real data of similar volume (for instance from the Internet) and test your methods. Provide references to the data source. How the running times observed compare with the running times observed on the synthesized data used in part 3? Can you draw conclusions from this comparison about the structure of the real data?