CS608 - Fall 2020 - Assignment #5

Assigned: October 30th, 2020
Due: November 9th, 2020 (Extra credit: December 1st, 2020)
NO LATE SUBMISSIONS.
Non-coding answers may be resubmitted once, after receiving feedback on the first attempt. **That is, if you do not submit anything in the first attempt, you cannot resubmit.**

Collaboration policy: The goal of assignment 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 study group is unable to solve a problem, it is your responsibility to get help from the instructor before the assignment is due.

For this assignment, you can form a team of up to three members. Each team must write up each problem solution and/or code any programming assignment without external assistance, even if you collaborate with others outside your team for discussions. You are asked to identify your collaborators outside your team. **If you did not work with anyone outside your team, 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 any member of the team cannot orally explain to the instructor.** No other student or team 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.

Submission Guidelines: Your team has to submit your work on Blackboard (no email) by the due date. Only one submission per team is necessary. For each class in the programming assignments you must use the header template provided in Blackboard. Make sure that you identify your team members in the header, and any collaborators outside your team, if none, write "none". Your code must follow the Java formatting standards posted in Blackboard. Format will also be part of your grade. To complete the submission, you have to upload two files to Blackboard: your source file and your class file. Your answers to questions that do not require coding must be included in the remarks section of the header. **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. Convoluted 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 #5 Grading Rubric

Coding:

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%		

Non-coding:
Embedded in questions.

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

Assignment:

We are given n points in the unit circle[1], $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for $i = 1, 2, \ldots, n$. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. The n points have to be sorted by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin. You can use the fragment of code below.

1. **20 points** Implement a method QuickSort() that sorts the points in $\Theta(n \log n)$ in expectation.

2. **30 points** Implement a method BucketSort() that sorts the points in $\Theta(n)$ in expectation. Be careful on how you define your buckets. Notice that uniform points in the circle is not the same as uniform points in the square.

3. **30 points** Write a method to test both sorting algorithms with different values of n. To produce the input points, implement the pseudocode in Algorithm 1. Fill the following table with the running times measured. Put the table in the code header. Adjust the values of n as needed to obtain at least four measurements.

	$n = 10^3$	$n = 10^4$	$n = 10^5$	$n = 10^6$
QuickSort				
BucketSort				

4. **20 points** According to your measurements, what is the running time of each sorting algorithm? Do they match the expected running time? If they do not, discuss what could be the reason.

[1] A circle of radius 1.

Algorithm 1: Pseudocode to draw a uniform point from a unit circle. The function random() must return a number drawn from $(0, 1]$ uniformly at random.

$t \leftarrow 2\pi \cdot \text{random}()$;
$u \leftarrow \text{random}() + \text{random}()$;
if $u > 1$ **then**
  $r \leftarrow 2 - u$
else
  $r \leftarrow u$
end
$(x, y) \leftarrow (r\cos(t), r\sin(t))$;