CPSC 331 (Spring 2023)
Assignment 5
Heaps and Graphs
(8% of final mark)
Due: Thursday June 15th at 11:59PM

Lead TA: The lead TA for this assignment is: Akram Ansari (mohdakram.ansari@ucalgary.ca)

Objective: Implement and compare two methods for All-Pairs-Shortest-Paths (APSP) in a Directed Graph

Use the RandomAdjMatrixGraph class provided on D2L to generate random, weighted, directed graphs for this assignment. The density is a factor from 1 to 10. For a graph with density 10, there is a chance that each pair of vertices will have more than one edge connecting them.

Implement Dijkstra's APSP algorithm presented in the slides. Also implement Floyd's ASAP presented in the lecture slides.

Part I
Implement both algorithms using an adjacency matrix representation of graphs in Java. Generate random graphs of size between $|V| = 500$ to 1000 with density 5 (medium). Compare the two algorithms by plotting the time needed (y-axis) to compute the APSP for each of these graph sizes (graph size is plotted on the x-axis).

Part II
Generate graphs of the same size ($|V| = 5000$) with varying density values between 1 and 9. Plot the running time of both methods from Part I, with the running time on the y-axis and the density of the graph on the x-axis.

Deliverables
- Java source code
- A PDF file containing a report (two pages maximum) that includes:
  - Your comments on both charts:
  - For the first chart, how do the two methods behave as the number of vertices increases? Why?
  - For the second chart, does the graph density affect the running time of these algorithms? If so, how? Does it affect one differently than the other? Why?

Submission: Submit the deliverables to the appropriate dropbox. The TA may provide extra submission requirements or instructions.

Collaboration: You are advised to work on this assignment in pairs. Groups of more than two members are not allowed. You may change your partner from the previous assignment. Any discussion with your colleagues outside your team regarding the assignment must be kept at a very high level.

Late Submission Policy: Late submissions will be penalized as follows:
-12.5% for each late day or portion of a day.

Academic Misconduct: Any similarities between submissions will be further investigated for academic misconduct. Your final submission must be your own team's original work. While you are encouraged to discuss the assignment with colleagues outside your team, this must be limited to conceptual and design decisions. Code sharing by any means with colleagues that are not in your team is prohibited. This includes and is not limited to looking at someone else's paper or screen. Any re-used code of excess of 5 lines must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

D2L Marks: Any marks posted on D2L are tentative and are subject to change (UP or DOWN).

Marking Rubric (90 points total)

Code	**Total 50 points**
Code compiles	5
Code runs	5
Dijkstra’s algorithm implementation	10
Floyd’s algorithm implementation	10
Experiment and produced values	10
Code is modular	5
Code contains enough documentation	5
Report	**Total 40 Points**
The charts are clear, readable, and convey the required message (10 each)	20
The discussion is thoughtful and utilizes the charts (10 each)	20

Submissions that do not compile will receive no points. Code that compiles but does not run will not receive more than 5/100.