

Due: 11:59pm ET Tuesday, November 14, 2023

Reminder

- No late work will be accepted without permission in advance from the instructor before the due date.
- All work submitted under your name must be your own. See Pace University's [Academic Integrity Code \(PDF\)](#).

Submission Instruction

- This assignment is an open-ended project about Graph Algorithms. You will work in a team of 3-4 people.
 - Submit your team formation (i.e., who is in the team) and the team name by 11:59pm ET Tuesday, October 31 at <https://classes.pace.edu> under **Assignment #3 Team Formation**.
 - Only one (1) person from each team needs to submit the team formation.
- The project consists of four (4) parts: 1) Software, 2) Documentation, 3) Peer Review, 4) **Optional** Extra Credit. **Follow instructions carefully.**
- Upload the file to the assignment section of the course homepage at <https://classes.pace.edu>.

Project Prompt

You are given two (2) files: airport.csv and route.csv as the data files:

- airport.csv contains the information about the airports listed in route.csv file.
- route.csv contains the information about the routes the airline represented is flying or helped by their partners for flying.

You will build a Java application that finds the shortest route to go from one airport to another. You do **NOT** need to consider the airports not listed in the data files.

- You must use at least one (1) graph algorithm to solve the problem.
- Feel free to make assumptions, and document them accordingly in your documentation file.

Hint: See <https://www.geeksforgeeks.org/program-distance-two-points-earth/> for the algorithm to calculate the distance between two (2) locations represented by latitude and longitude.

Part 1: Software [50 pts]

Instruction: You must compress all .java **source code files** into one (1) zip file; name your zip file: {Team_Name}.zip; only one (1) person in the team needs to upload the zip file. All your codes **MUST** compile and execute using the instructions in the documentation file. The codes that don't compile or execute will result in **zero (0) point**.

Below are the requirements your software must cover:

- A user must be able to input the departure airport (in 3-letter code) where a route will be searched from.
- A user must be able to input the arrival airport (in 3-letter code) where a route will be searched to.
- A user must be able to see the following information about the shortest route found: 1) full itinerary (a list of airports in 3-letter codes the route covers), 2) total distance of the route in kilometers (km).

Hint 1: See <https://www.geeksforgeeks.org/program-distance-two-points-earth/> for the algorithm to calculate the distance between two (2) locations represented by latitude and longitude

Hint 2: Total distance is calculated as the sum of distances for all flight segments involved. (i.e., if the full itinerary covers A – B – C, the total distance is calculated as *the distance between A and B + the distance between B and C*.)

Part 2: Documentation [30 pts]

Instruction: Name your file: {Team_Name}.docx; only one (1) person in the team needs to upload the file.

In software engineering, the documentation is the beginning and end of the process. At the beginning of the project, you will have to document the software design decision you are making. At the end of the project, you will document how your code is supposed to work (i.e., how to compile the software, how to use the software, and what gotcha's the users need to watch out for). See below for the **suggested** information; feel free to add additional information:

- How are airports and routes are modeled and implemented?
- How is the chosen graph algorithm used to achieve the requirements listed in Part 1?
- How should a user compile your codes and execute your software?

Part 3: Peer Review [20 pts]

Instruction: Download '2023fa_cs608_peer_review.docx' file from Assignment #3 section, and fill in the table inside the file. **Everyone** must upload the file.

Reminder:

1. Grade your team members in 2 criteria: Participation and Quality out of **10** in a whole number.
2. For each criterion (participation and quality) respectively, each team member must be assigned a different grade (**i.e., no two team members can have the same grades assigned**).

Part 4: Optional Extra Credit [30 pts]

Instruction: You may implement the following three (3) additional requirements in addition to the software implemented in Part 1. **Document** which of the following extra credit requirements in the file written in Part 2 explicitly. You may earn **up to** 30 points in Assignment #3, which is **up to 3%** in your final grade.

Below are the additional requirements your software **MAY** cover:

- A user can see the airport names in the full itinerary.
- A user can provide the prefix of the 3-letter airport codes and search available airports when inputting departure and arrival airports. (i.e., a user can type in 'A' to see all the airports whose 3-letter codes start with 'A', and select one airport from the list).
- A user can choose to **exclude** the routes flown by 'Regional' operators; if a route is flown by 'Regional' and other operators, a route can still be included in the final itinerary.