

Due: 11:59pm ET Thursday, April 13, 2023**Reminder**

- **IMPORTANT:** No late work will be accepted without permission in advance from the instructor before the due date. Everyone has 1 (one) chance during the semester to submit homework late.
- All work submitted under your name must be your own. See Pace University's [Academic Integrity Code \(PDF\)](#).

Instruction

- This assignment is an open-ended project about Graph Algorithms. You will work in a team of 2-3 people.
 - Submit your team formation (i.e., who is in the team) and the team name by 11:59pm ET Friday, March 31, 2023 at <https://classes.pace.edu> under Assignment #4 Team Formation.
 - 2-people team will receive 10% boost In your **adjusted score** of the assignment.
- The project consists of 2 parts: 1) documentation, 2) software. Follow the instructions carefully.
- Upload the file to the assignment section of the course homepage at <https://classes.pace.edu>.

Project Prompt

You will build a Java application that can find the fastest route to go from one MTA subway station to another using New York City subway system. 'mta_stations.csv' file contains the information about each subway station in MTA system.

- You must use at least one (1) graph algorithm to solve the problem.
- Feel free to define what “the fastest route” means based on the data file you are provided.
- Feel free to make assumptions, and document them accordingly in your documentation file.

Part 1: Documentation [40 pts]

Your documentation must be named in the format: {Team Name}.docx.

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. During the project, you will document what has changed from the original design of the software during the implementation. At the end of the project, you will document how your code is supposed to work (i.e., how to build the software, how to use the software, and what gotcha's the users need to watch out for). Below is the list of **suggested information** you may want to put in your documentation for this project.

- What “use cases” (see Part 2) are you supporting in your software?
- How will you build a “graph” of New York City subway stations based on the csv file provided?
- How are you defining “the fastest route” based on the information provided?
- How will you model the transfer between the train lines? Does it affect how you calculate “the fastest route”?
- What assumptions are you making in building this software?
- What “Classes” and “Methods” are built in your software? What algorithm did you use to find “the fastest route”?
- How is your software compiled, and how should it be executed?

Part 2: Software [60 pts]

Your software must be zipped in the format: {Team Name}.zip. All your codes MUST compile using the instruction in the documentation file.

Below are the use cases your software need to cover. P0 means “Priority 0”; these use cases must be supported in your software. P1, P2 mean “Priority 1”, “Priority 2”; these use cases are bonus use cases for extra credits; tackle P1 use cases first before P2 use cases.

P0	A user is able to input the start station and the end station.
P0	A user is able to see what trains they need to take to complete the routes.
P0	A user is able to see at which stations they need to transfer to complete the routes.
P0	A user is able to see the results where the “transfer time” is appropriately captured.
P1	A user is able to input the day/time they are taking the subway and see different results, if necessary.
P1	A user is able to type in the partial name of the train stations and search the stations they want to select.
P2	A user is able to see how long it will take to take “the fastest route” (Note: you will need to make appropriate assumptions to calculate the amount of time that will need to take the routes).
P2	A user is able to run the software using a graphical user interface (GUI).