CME2201 ASSIGNMENT-2

In this assignment, you are expected to compute electric current flow of a power distribution plant.

# Scenario

Suppose that nodes (Figure 1) represent power distribution plants and weighted edges represent the maximum power of electric current that can be flowed through the given direction at a particular time.

What is the maximum power of electric current that can be sent from the plant A to the plant C in a day? At beginning of the day, the plant A can deliver maximum 18 kwatt power as the sum of outgoing edges’ weights (9+2+7=18). In the next step, plant B, D, and G should flow the electric current they have received. The plant B can send 4 kwatt power to E, 2 kwatt power to D, while the rest 3 kwatt power will be absorbed, since there is not enough capacity to transfer this power via the outgoing edges of B. This process will continue until the destination plant C receives all possible electric power via its incoming edges.

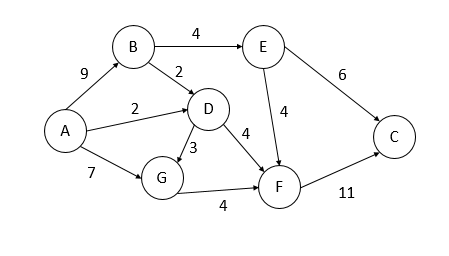
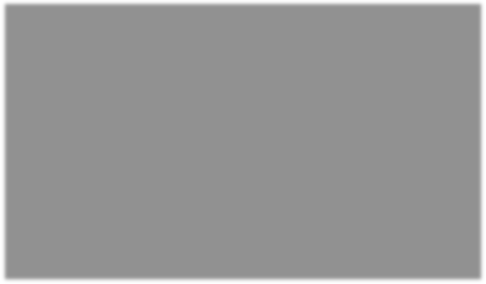


Figure 1: A sample electric current flow graph for power distribution plants.

# What is expected from your code?

You are given a ‘graph.txt’ file which contains a directed graph in the edge-list format. This file contains three columns as ‘source node’, ‘destination node’ and ‘edge-weight in between’ as a tab delimited format. You are expected to create a directed graph whose directions are constructed from ‘source’ to ‘destination’ with the given ‘edge-weight’ value.

Then, your program must ask to the user for a source and a destination plant name. Your code will run to find the maximum number of electric current power that can flow from source to destination plants provided by the user.

# Main Tasks:

1. What is the maximum power of electric current that can be flowed from source to destination in one day?
2. Which edges’ capacity should be increased at first to increase the amount of received power of electric current at the destination plant?
3. How much should we maximize these edges found in the task 2?

# Rules:

You are not allowed to use built-in Java libraries in the creation of the graph and in implementation of main tasks. You can use only **arraylist** and **hash-map** data structure in Java libraries if needed. Usage of adjacency matrix is suggested.

If you get a code from the internet and update it, please state the site you received in the comment line at the beginning of the received code.

# Due date: August 16, 2022, 23:55

**Control (demo) dates:** August 17 and 18, 2022.

# Submission

You must upload your all ‘.java’ files as an archive file (.zip or .rar) to the Sakai platform. Your archived file should be named as ‘studentnumber\_name\_surname.rar.zip’, e.g., 2022510999\_Ali\_Cuvitoglu.rar.

Incorrect naming will cause you get a lower score!

Prepare and upload a report only if your code is not completed. Explain the parts of your code that don't run (or missing parts) and why it doesn't run. Don’t forget to add your references.

# Plagiarism Control

The submissions will be checked for code similarity. Highly similar assignments will be graded as zero, and they will be announced in the Sakai.

# Grading Policy

|  |  |
| --- | --- |
| **Tasks** | **Points (out of 100)** |
| Implementation of Graph creation (read file and insert into matrix) | 20 |
| Main Task-1 | 30 |
| Main Task-2 | 30 |
| Main Task-3 | 20 |