TCP2101 Algorithm Design & Analysis

Session 1 2020/2021

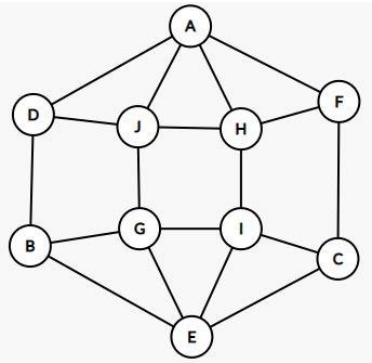
Assignment 2

Instructions

- This is a group assignment with **3 persons per group**.
- Each group will implement the assignment using **C++ language**.
- You are required to compile and run the program to generate the specified sets of data, and use the data to demonstrate the programs.
- Then write the programs to demonstrate the algorithms and record the results of processing the data sets.
- Write a report to describe the algorithms implemented, results of the tests, discussions/reasoning for the results obtain and a conclusion on the algorithms (e.g. suitability, possible improvements etc.)
- All members must present to explain the algorithms, demo the program, present the experimental results, conclude the findings, and Q&A. The presentation date, time and venue will be announced at MMLS/Google classroom.

Scenario

- You are assigned to assist the captain of a spaceship who collects items/products from a number of planets in the Beta Quadrant. The routes that is followed by the spaceship is represented by the following 'map':



- The data of the planets can be generated using a given C++ program. You will enter 3 student IDs (group members) and the program will generate a list of planets with the set of values for coordinates (x, y, x), item weight and item profit (for the item to be collected from each planet).

- The home planet is Planet A, and you will analyse the data to produce the following:
  - List of sorted edges according to distance
  - List of sorted planets according to value
  - The shortest path to each planet
  - The minimum spanning tree connecting all the planets
  - The optimal set of planets to visit before returning home

Data Sets

- Compile and run the C++ program to generate your data set for the 10 planets.
- Enter the 3 student IDs of group members and the program will generate the data for the 10 planets (Planets A-J) into a file named "A2planets.txt".
- Each row in the file will contain the data for a planet. The data consists of planet name, x-coordinate, y-coordinate, z-coordinate, weight, and profit. The weight and profit refers to the item to be collected from the planet.
- For example:

  Planet_C 123 572 396 15 30

- Use the following formula to calculate the distance between planets:

$$distance_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$$

- All your programs will read this file and process the data to generate the outputs.

Programs

Write the following programs:

a. Program 1 – Display and Sort. Read the data set, calculate the distances of connected planets, then create an adjacency matrix and an adjacency list using the distances as 'weights' of the edges. Display the adjacency matrix and adjacency list. Finally use Merge-Sort or Quick-Sort to produce the following lists:
   - List of edges in ascending order of distance
   - List of planets in descending order of value.

b. Program 2 – Shortest Paths. Write a program to identify the shortest paths from Planet A to the other planets using Dijkstra's Algorithm. Then display the shortest distance to each planet. Finally draw the graph representing the shortest paths.

c. Program 3 – Minimum Spanning Tree. Write a program to identify the Minimum Spanning Tree using Kruskal's Algorithm. Display the edges of the tree and draw the graph representing the Minimum Spanning Tree.

d. Program 4 – Dynamic Programming. Your spaceship has the capacity of 80 tons. Write a program to identify the set of planets to visit without returning to home planet, using the 0/1 Knapsack Algorithm (assume that if you reach a planet, you have to take all the items/products). Display the resulting matrix and the list of planets to visit, with the weights and benefits from each planet.

For each program, describe the main program and algorithms in the report.

Deliverables

- Report containing
  - Data generation – description of data set
  - Program 1 implementation – main program and algorithms, and the program outputs.
  - Program 2 implementation - main program and algorithms, and the program outputs.
  - Program 3 implementation - main program and algorithms, and the program outputs.
  - Program 4 implementation - main program and algorithms, and the program outputs.

- Programs (Executable, Program Listing & Relevant System Files)
  - Submit the project folders (if using Code::Blocks or any IDE (specify)

- Presentation
  - Each member explains a program. All members must be able to explain any programs implemented in the group.

Deadlines

- Report Submission: Sunday 28 February 2021

- Presentation: Week 15 (1 – 7 March 2021)

Assessments

Mark Distribution	Submission Type	Descriptive Elements	Weightage	Rate (0-5)	Total (Weightage * Rate)
Cognitive and Affective Components	Report 1	1) Program 1 implementation and results	1		
		2) Program 2 implementation and results	1		
		3) Program 3 implementation and results	1		
		4) Program 4 implementation and results	1		
		5) Report contents and quality	1		
	Presentation 1	1) Demonstration & Explanation	1		
		Report 1		**Max (25%)**	
		Presentation 1		**Max (5%)**	

COGNITIVE COMPONENT RUBRIC
PROJECT II – DESIGN / ARCHITECTURE / INTERFACES / DATABASE

Descriptive Elements	Very Weak (1)	Weak(2)	Fair(3)	Good(4)	Excellent(5)
1) Program 1 implementation and results	Program implemented poorly and minimal results produced.	Program implemented with defects and some results produced.	Program implemented acceptably and sufficient results produced.	Program implemented well and results produced quite efficiently	Program implemented effectively and results produced efficiently
2) Program 2 implementation and results	Program implemented poorly and minimal results produced.	Program implemented with defects and some results produced.	Program implemented acceptably and sufficient results produced.	Program implemented well and results produced quite efficiently	Program implemented effectively and results produced efficiently
3) Program 3 implementation and results	Program implemented poorly and minimal results produced.	Program implemented with defects and some results produced.	Program implemented acceptably and sufficient results produced.	Program implemented well and results produced quite efficiently	Program implemented effectively and results produced efficiently
4) Program 4 implementation and results	Program implemented poorly and minimal results produced.	Program implemented with defects and some results produced.	Program implemented acceptably and sufficient results produced.	Program implemented well and results produced quite efficiently	Program implemented effectively and results produced efficiently
5) Report contents and quality	Report is poorly written and is difficult to understand	Report is written with limited clarify and coherence and require further improvements	Report is written fairly coherently and clearly but require minor improvements	Report is written coherently and clearly	Report is well written with excellent coherence and clarity

Note for Rate: 0 = no submission

AFFECTIVE COMPONENT RUBRIC
PRESENTATION 1

Descriptive Elements	Very Weak (1)	Weak(2)	Fair(3)	Good(4)	Excellent(5)
1) Demonstration & Explanation	Not able to demonstrate and explain clearly and require major improvements	Able to demonstrate and explain but require further improvements	Able to demonstrate and explain fairly clearly and require minor improvements	Able to demonstrate and explain clearly	Able to demonstrate and explain with great clarity

Note for Rate: 0 = no submission