CCP6214: ALGORITHM DESIGN AND ANALYSIS TERM 2410

This assessment is the fulfilment of the following course learning objective of CCP6214: Algorithm Design and Analysis:

CLO3	Write efficient algorithms in terms of space and time complexities (Cognitive level:6)

	Instruction Section
1	This is a group assignment with the **minimum** of **3** persons to **maximum 4** persons per group.
2	**Deadlines** • Assignment deliverables submission: **14 June 2024 (Friday), 11:59pm** • Presentation**: Week 13 (17 June – 21 June 2024) – Week 14 (24 June – 28 June 2024)**
3	Implement the assignment using **C++** language or other programming languages that you (all group members) are confident with. Implementation using other languages is up to own discretion. The code should be able to generate the sample set of data described within question descriptions and implement algorithms accordingly.
4	Write a **report** to describe the algorithms implemented, step-by-step algorithm illustrations, results of the experiments, screenshots, code parts, discussions with reasoning for the results obtained and a conclusion on the algorithms such as suitability, possible improvements etc. Include all the citations and references in APA style in your report on the final page of the report.
5	Group leader shall zip the submission (one **report** in docx / pdf, coding or **cpp** file(s) in respective folders for each question, input files, output files, one presentation slides file). Start the assignment as soon as possible. **No extension of submission deadline** will be made. To make testing easier and save time during presentation, your program shall never clear the computer screen and store output to output file(s) such as txt files.
6	You are required to follow algorithms and formats based on the given dataset generation. For first dataset generation, the seed number is group leader student id. For second dataset generation, the seed number is summation of members' student id.
7	**All members** must **present** to explain the algorithms, demonstrate the working programs, present the experimental results, conclude the findings, and etc. No mark will be given to the sleeping member or any instruction is not followed. Book the interview date, time and venue with your lecturer. There will be an **interview** and **Q&A session** for the assignment to validate your work.
8	Assessment criterions stated in marking rubrics section are applicable to evaluation of all deliverables: **report**, **codes, presentation slides** and **presentation session (interview** and **Q&A)**.
9	You are reminded that **no plagiarism** is allowed in any means. If any plagiarism is found, **zero** mark will be awarded. Do not share your code with anyone. If detected, all groups involved will be penalized with zero marks.

	Recommendation Section
1	You are to **distribute** the programming tasks among the group members and meet at least 2 times a week to report your **progress** and **discuss** the problems you are facing with other group members. You can try to divide the programming tasks accordingly: • Algorithm 1 (dataset 1 and 2 generation) implementation and reporting and other questions • Algorithm 2 (Heap and Selection Sort) implementation and reporting and other questions • Algorithm 3 (Dijkstra's and Kruskal's) implementation and reporting and other questions • Algorithm 4 (Dynamic Programming) implementation and reporting and other questions
2	You are also advised to integrate your program from the beginning and only share the bug-free program with other members whenever you make changes to the program.
3	You are also advised to **keep** your program version properly and store your **backup** somewhere safe (i.e., pen drive or external drive or cloud).
4	You should **add comments** whenever possible describing the **purpose** of a variable or a block of code. A properly commented program can make the maintenance easy and readable.

Assignment Questions Section

1 **Dataset 1** (For implementation in Q2)

Write an algorithm to generate 6 sets of data with random numeric seeding. Use group leader's ID as the random seed reference to generate dataset of size(s) as follows:

Set 1: 100
Set 2: 1,000
Set 3: 10,000
Set 4: 100,000
Set 5: 500,000
Set 6: 1,000,000

Example of usage of group leader's ID as random seed reference (e.g. ID 1234000001)
Dataset 1:

423	100	101	123	321	32	13	14	222	…

Store random seeded dataset(s)/input(s) in e.g. txt file(s).

Dataset 2 (For implementation in Q3 and Q4)

You are assigned to assist in designing locations of stars with values of each star should hold for a star conquer quest. In order to ensure the feasibility for participants to win the conquer quest with designed stars' locations, theoretical analyses need to be conducted.

Write an algorithm to generate dataset for 20 locations of stars (vertices) with 54 routes (edges) that connect between the stars. Each star needs to at least connect to 3 other stars. Use the sum of other group members' ID number to generate data consisting of values for star name, x-coordinate, y-coordinate, z-coordinate, weight and profit. The weight and profit refer to the points to be collected from conquering the star.

Example of usage of sum of other group members' ID number as random seed reference (e.g. 1234000002 +1234000003 + 1234000004 = 3702000009)
Dataset 2:

Name	x	y	z	weight	profit
Star A	370	9	72	23	90

***Example of usage student ID that has letters within the ID, as random seed reference (e.g. 241UC2407F)*
Refer to ASCII value for each letter in student ID and take the last digit: U = 085, C = 067, F = 070
Thus, the student ID 241UC2407F, will be 2415724070

Use the following equation to calculate the distance between connected stars:

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

Draw the designed stars with routes with details of star(s)' name as example shown in Figure 1:

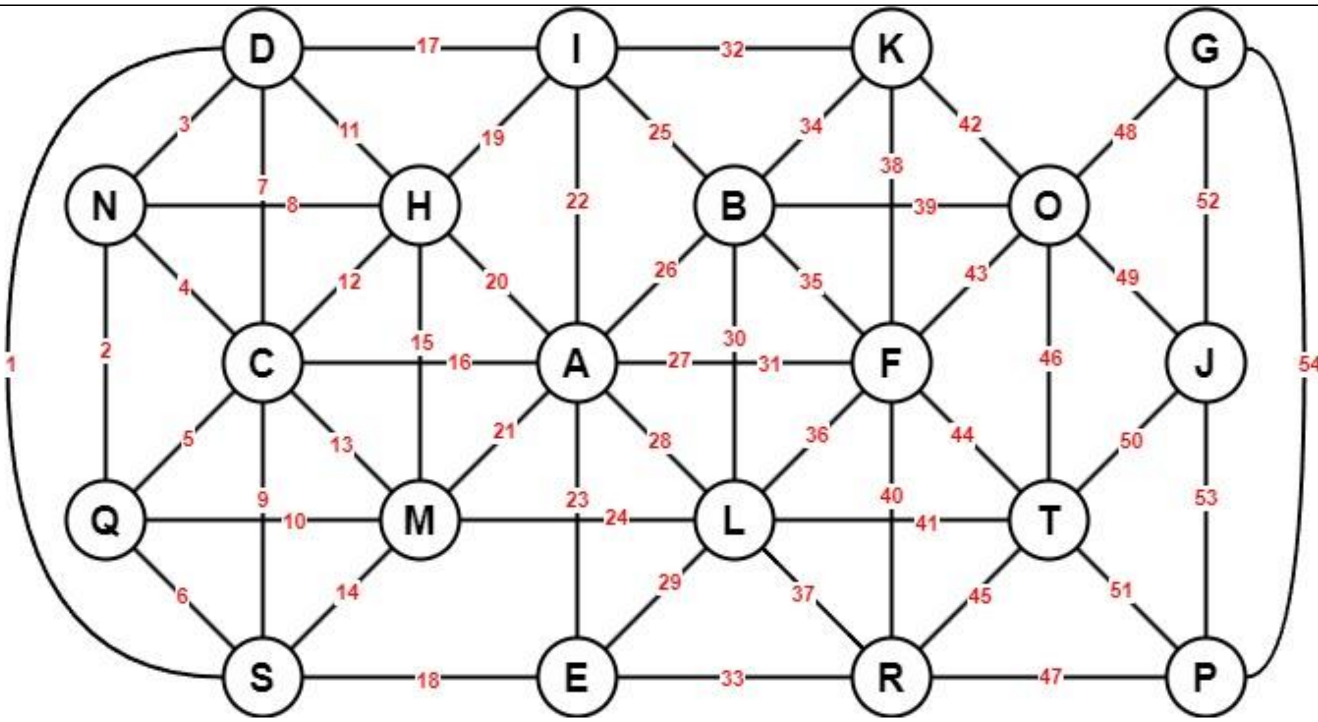
<table>
<tr><td></td><td>


Figure 1: Example of star location design with routes.

Store random seeded dataset(s)/input(s) in e.g. txt file(s).
</td></tr>
<tr><td>2</td><td>
Use Dataset 1 for implementation of Heap Sort and Selection Sort Algorithms.
Heap Sort
Store the data in a priority queue using a heap. Store sorted dataset(s)/output(s) in e.g. txt file(s). Record the time to insert all data into the priority queue. Dequeue the data to test the timing of the priority queue. Plot the graph of timing vs dataset size and discuss about the time and space complexity of the algorithm.
Selection Sort
Sort the dataset(s) using selection sort. Store sorted dataset(s)/output(s) in e.g. txt file(s). Record the time to complete sorting of data. Plot the graph of timing vs dataset size and discuss about the time and space complexity of the algorithm.
</td></tr>
<tr><td>3</td><td>
Use Dataset 2 for the implementation of Dijkstra's and Kruskal's Algorithm.
Shortest Paths
Write a program to identify the shortest paths from Star A to the other stars using Dijkstra's Algorithm. Then display the shortest distance to each star (store the output(s) in e.g. txt file(s)). Draw the graph representing the shortest paths. Discuss about the time and space complexity of the algorithm.
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 (store the output(s) in e.g. txt file(s)). Discuss about the time and space complexity of the algorithm.
</td></tr>
<tr><td>4</td><td>
Use Dataset 2 for the implementation of 0/1 Knapsack Algorithm.
Dynamic Programming
Assume that each participant travels in a spaceship which have maximum capacity of conquering 800kg of stars. Write a program to identify the set of star(s) to visit without returning to beginning star location using the 0/1 Knapsack Algorithm (assume that once the participant reach a star, participant has to conquer and plunder the full weight of that star). Draft out the resulting matrix and the list of stars to visit, with the weights and profits from each star to be winner of the star conquer quest (store the output(s) in e.g. txt file(s)). Discuss about the time and space complexity of the algorithm.
</td></tr>
</table>

<table>
<tr><th colspan="2">Deliverables Section</th></tr>
<tr><td></td><td>
Report
Cover page consisting a table which remarks each member's contribution. Example of the table as follows:
</td></tr>
</table>

Num	Student ID	Student Name	Task descriptions	Percentage %
1	1234000001	Mark Kelly (group leader)	Q1: Dataset 1 and 2	25
2	1234000002	Norlan Jimmie	Q2: Heap and Merge Sort	25
3	1234000003	Joey Lane	Q3: Dijkstra’s and Kruskal	25
4	1234000004	Rita Bane	Q4: 0/1 Knapsack	25

- Dataset generation for Dataset 1 and Dataset 2 – algorithms and programs.
- Heap Sort and Selection Sort – algorithm, programs for algorithm implementation, experimental results and discussions.
- Shortest Paths and Minimum Spanning Tree – algorithm, programs for algorithm implementation, experimental results and discussions.
- Dynamic Programming – algorithm, programs for algorithm implementation, experimental results and discussions.

Programs (Input and Output files, Executables, Program Listing and Relevant System Files)

- Compressed (zipped) coding project folder

Presentation

- Presentation slides
- Each member to present a question. All members must be able to explain how the algorithms were implemented and experimental results.
- Q&A and Interview is applicable within presentation session. Each member is expected to be able to heed at least one of the questions addressed by lecturer.
- In any case of student could not address any questions by lecturer or provided vague / questionable answers during presentation, but produced correct reporting and codes, further investigation of plagiarism will be conducted by the lecturer.

Assessments Evaluation Section				
Components	Elements	Weight	Rate (0-5)	Total
Report, Programs and Presentation	Question 1: Dataset 1	5		
	Question 1: Dataset 2	5		
	Question 2: Heap Sort	5		
	Question 2: Selection Sort	5		
	Question 3: Shortest Path	5		
	Question 3: Minimum Spanning Tree	5		
	Question 4: Dynamic Programming	5		
Individual	Elaborations, Interview and Q&A session	5		
			Total (40%)	

Rubrics Section						
Criteria	No Attempt (0)	Very Weak (1)	Weak (2)	Average (3)	Good (4)	Excellent (5)
Part 1 dataset generation algorithm, implementations and discussions	No attempt made or implementation is similar with other group(s) with no justification(s).	Algorithm implemented poorly with minimal datasets produced and most of the explanations were not of own words. Implied lack of ability to identify rightful algorithm as solution.	Algorithm implemented with major defects and some datasets were produced with brief explanations of own words. Implied lack of ability to depict rightful algorithm as solution.	Algorithm implemented with minor defects and some datasets were produced with sufficient explanations of own words. Implied lack of ability to manipulate rightful algorithm as solution.	Algorithm implemented without defect and datasets were produced with good explanations accompanied with own ideas. Implied ability of designing rightful algorithm as solution.	Algorithm implemented effectively and datasets were produced with detailed explanations accompanied by own ideas or examples. Implied ability of constructing rightful algorithm as solution.
Part 2 dataset generation algorithm, implementations. station design and discussions	No attempt made or implementation and station design are similar with other group(s) with no justification(s).	Algorithm implemented poorly or resultant station design similar to other group(s) with justifications and most of the explanations were not of own words. Implied lack of ability to identify rightful algorithm as solution.	Algorithm implemented with major defects and minimal station design produced with brief explanations of own words. Implied lack of ability to depict rightful algorithm as solution.	Algorithm implemented with minor defects and sufficient station design produced with acceptable explanations of own words. Implied lack of ability to manipulate rightful algorithm as solution.	Algorithm implemented without defect and station design is unique with good explanations accompanied with own ideas. Implied ability of designing rightful algorithm as solution.	Algorithm implemented effectively and unique station design is produced with detailed explanations accompanied by own ideas or examples. Implied ability of constructing rightful algorithm as solution.
Heap Sort implementation, results and discussions	No attempt made or implementation, experimental results and discussions are similar with other group(s), with no justification(s).	Algorithm implemented poorly with minimal results produced and most of the explanations were not of own words. Implied lack of ability to reproduce from learnt concepts.	Algorithm implemented with major defects and some results were produced with brief explanations of own words. Implied lack of ability to comprehend learnt concepts.	Algorithm implemented with minor defects and sufficient results were produced with acceptable explanations of own words. Implied lack of ability to operate learnt concepts for real-world solutions.	Algorithm implemented without defect and results were produced with good explanations accompanied with own ideas. Implied ability of associating learnt concepts for real-world solutions.	Algorithm implemented effectively and results were produced with detailed explanations accompanied by own ideas or examples. Implied ability of integrating learnt concepts for real-world solutions.
Selectiom Sort implementation,	No attempt made or implementation,	Algorithm implemented poorly	Algorithm implemented with	Algorithm implemented with	Algorithm implemented without	Algorithm implemented

results and discussions	experimental results and discussions are similar with other group(s), with no justification(s).	with minimal results produced and most of the explanations were not of own words. Implied lack of ability to reproduce from learnt concepts.	major defects and some results were produced with brief explanations of own words. Implied lack of ability to comprehend learnt concepts.	minor defects and sufficient results were produced with acceptable explanations of own words. Implied lack of ability to operate learnt concepts for real-world solutions.	defect and results were produced with good explanations accompanied with own ideas. Implied ability of associating learnt concepts for real-world solutions.	effectively and results were produced with detailed explanations accompanied by own ideas or examples. Implied ability of integrating learnt concepts for real-world solutions.
Shortest path implementation, results and discussions	No attempt made or implementation, experimental results and discussions are similar with other group(s), with no justification(s).	Algorithm implemented poorly with minimal results produced and most of the explanations were not of own words. Implied lack of ability to reproduce from learnt concepts.	Algorithm implemented with major defects and some results were produced with brief explanations of own words. Implied lack of ability to comprehend learnt concepts.	Algorithm implemented with minor defects and sufficient results were produced with acceptable explanations of own words. Implied lack of ability to operate learnt concepts for real-world solutions.	Algorithm implemented without defect and results were produced with good explanations accompanied with own ideas. Implied ability of associating learnt concepts for real-world solutions.	Algorithm implemented effectively and results were produced with detailed explanations accompanied by own ideas or examples. Implied ability of integrating learnt concepts for real-world solutions.
Minimum spanning tree implementation, results and discussions	No attempt made or implementation, experimental results and discussions are similar with other group(s), with no justification(s).	Algorithm implemented poorly with minimal results produced and most of the explanations were not of own words. Implied lack of ability to reproduce from learnt concepts.	Algorithm implemented with major defects and some results were produced with brief explanations of own words. Implied lack of ability to comprehend learnt concepts.	Algorithm implemented with minor defects and sufficient results were produced with acceptable explanations of own words. Implied lack of ability to operate learnt concepts for real-world solutions.	Algorithm implemented without defect and results were produced with good explanations accompanied with own ideas. Implied ability of associating learnt concepts for real-world solutions.	Algorithm implemented effectively and results were produced with detailed explanations accompanied by own ideas or examples. Implied ability of integrating learnt concepts for real-world solutions.
Dynamic programming implementation, results and discussions	No attempt made or implementation, experimental results and discussions are similar with other group(s), with no justification(s).	Algorithm implemented poorly with minimal results produced and most of the explanations were not of own words. Implied lack of ability to reproduce from learnt concepts.	Algorithm implemented with major defects and some results were produced with brief explanations of own words. Implied lack of ability to comprehend learnt concepts.	Algorithm implemented with minor defects and sufficient results were produced with acceptable explanations of own words. Implied lack of ability to operate	Algorithm implemented without defect and results were produced with good explanations accompanied with own ideas. Implied ability of associating learnt concepts for	Algorithm implemented effectively and results were produced with detailed explanations accompanied by own ideas or examples. Implied ability of integrating learnt

				learnt concepts for real-world solutions.	real-world solutions.	concepts for real-world solutions.
Individual: Elaborations, Interview and Q&A session	No attempt made or reading from slide or report during presentation and did not present own inputs during Q&A session. Implied no understanding on completed work.	Redundant explanations without highlighting main discussion points in presentation and addressing questioned assignment area. Implied lack of ability to relate implemented algorithm with learnt concepts.	Brief explanations with brief highlight of main discussion points in presentation and when addressing further questions in assignment area. Implied lack of ability to illustrate learnt concepts.	Moderate explanations with acceptable highlights of main discussion points in presentation and when addressing further questions of assignment area. Implied lack of ability to demonstrate learnt concepts.	Good explanations with highlights of main discussion points in presentation and when addressing further questions of assignment area. Implied ability of associating learnt concepts to be part of solution of real problems raised.	Detailed explanations, creatively highlighting main discussion points in presentation and when addressing further questions of assignment area. Implied ability of integrating learnt concepts to propose solutions to real problems raised.