

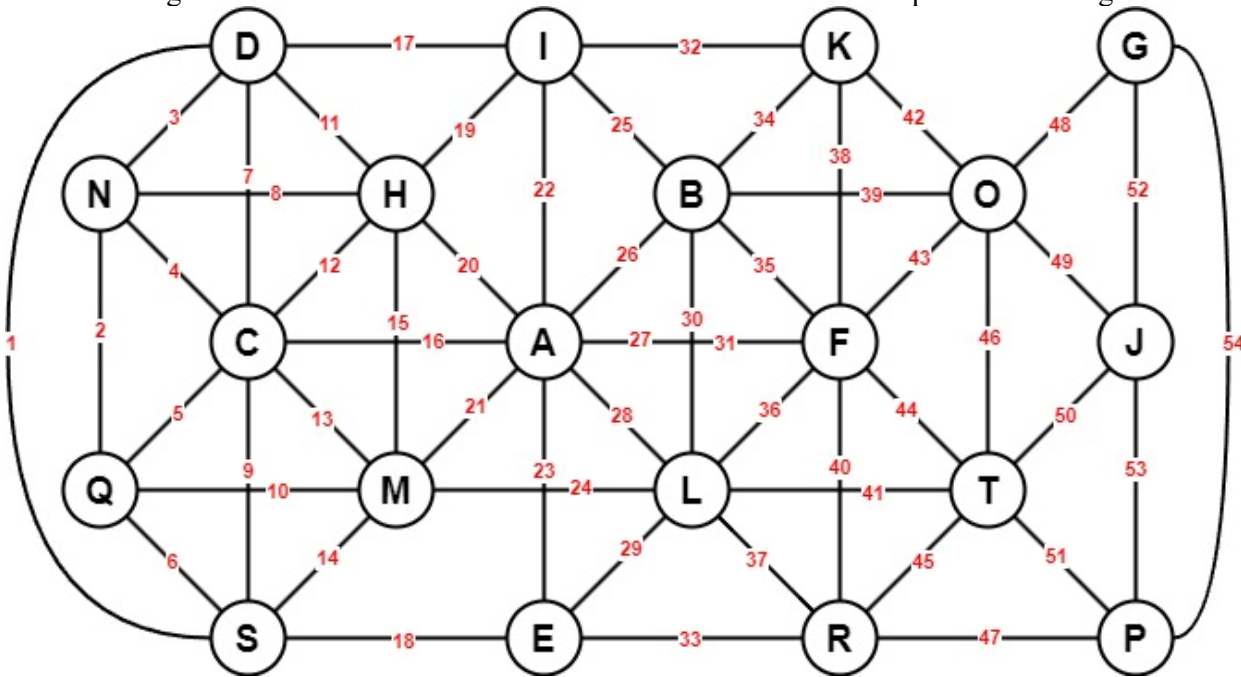
This assessment is the fulfilment of the following course learning objective of TCP2101: Algorithm Design and Analysis:

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 <ul style="list-style-type: none"> Assignment deliverables submission: 26 January 2024 (Friday), 11:59pm Presentation: Week 13 (29 Jan – 2 Feb 2024) – Week 14 (4 Feb – 8 Feb 2024)
3	Implement the assignment using C++ language. 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, coding or cpp file(s) in respective folders for each question, input files, output files, one mp4 video 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 output to the files.
6	You are required to follow algorithms and formats based on the given dataset generation. For all dataset generation, the seed number is your group leader 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	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 3 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: <ul style="list-style-type: none"> Algorithm 1 (dataset 1 and 2 generation) implementation and reporting and other questions Algorithm 2 (Heap and Merge 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	<p>Dataset 1 Write an algorithm to generate 4 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 2: 10,000 Set 3: 100,000 Set 4: 500,000</p> <p>Dataset 2 You are assigned to assist in designing locations of stations with treasures that each station should hold for a treasure hunt project. In order to ensure the feasibility for participants to win the treasure hunt with designed stations' locations, a theoretical analysis need to be conducted. Write an algorithm to generate dataset for 20 stations (vertices) with 54 routes (edges) that connect between stations. Each station needs to at least connect to 3 stations. Use the sum of other group members' ID number to generate data consisting of values for station name, x-coordinate, y-coordinate, z-coordinate, weight and profit. The weight and profit refer to the treasure to be collected from the station. For example: Station A 265 324 427 18 40 Use the following equation to calculate the distance between stations:</p> $distance_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$ <p>Draw the designed stations with routes with details of station name as example shown in Figure 1:</p>  <p>Figure 1: Example of station design with routes.</p>
2	<p>Use Dataset 1 for implementation of Heap Sort and Merge Sort Algorithm.</p> <p>Heap Sort Store the data in a priority queue using a heap. 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.</p> <p>Merge Sort Sort the dataset(s) using merge sort. 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.</p>
3	Use Dataset 2 for the implementation of Dijkstra's and Kruskal's Algorithm.

	<p>Shortest Paths Write a program to identify the shortest paths from Station A to the other stations using Dijkstra's Algorithm. Then display the shortest distance to each station. Draw the graph representing the shortest paths. Discuss about the time and space complexity of the algorithm.</p> <p>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. Discuss about the time and space complexity of the algorithm.</p>
4	<p><i>Use Dataset 2 for the implementation of 0/1 Knapsack Algorithm.</i></p> <p>Dynamic Programming Assume that each participant travels in a truck which have maximum capacity of 800kg. Write a program to identify the set of station to visit without returning to home station using the 0/1 Knapsack Algorithm (assume that if you reach a station, you have to take all the treasure at that station). Draft out the resulting matrix and the list of stations to visit, with the weights and profits from each station to be winner of the treasure hunt project. Discuss about the time and space complexity of the algorithm.</p>

Deliverables Section

Report	<ul style="list-style-type: none">Cover page consisting a table which remarks each member's contribution. Example of the table as follows:<table><tr><th>Num</th><th>Student ID</th><th>Student Name</th><th>Task descriptions</th><th>Percentage %</th></tr><tr><td>1</td><td>100000001</td><td>Mark Kelly</td><td>Q1: Dataset 1 and 2</td><td>25</td></tr><tr><td>2</td><td>100000002</td><td>Norlan Jimmie</td><td>Q2: Heap and Merge Sort</td><td>25</td></tr><tr><td>3</td><td>100000003</td><td>Joey Lane</td><td>Q3: Dijkstra's and Kruskal</td><td>25</td></tr><tr><td>4</td><td>100000004</td><td>Rita Bane</td><td>Q4: 0/1 Knapsack</td><td>25</td></tr></table>Dataset generation for Dataset 1 and Dataset 2 – algorithms and programs.Heap Sort and Merge 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.	Num	Student ID	Student Name	Task descriptions	Percentage %	1	100000001	Mark Kelly	Q1: Dataset 1 and 2	25	2	100000002	Norlan Jimmie	Q2: Heap and Merge Sort	25	3	100000003	Joey Lane	Q3: Dijkstra's and Kruskal	25	4	100000004	Rita Bane	Q4: 0/1 Knapsack	25
Num	Student ID	Student Name	Task descriptions	Percentage %																						
1	100000001	Mark Kelly	Q1: Dataset 1 and 2	25																						
2	100000002	Norlan Jimmie	Q2: Heap and Merge Sort	25																						
3	100000003	Joey Lane	Q3: Dijkstra's and Kruskal	25																						
4	100000004	Rita Bane	Q4: 0/1 Knapsack	25																						
	Programs (Executables, Program Listing and Relevant System Files) <ul style="list-style-type: none">Compressed (zipped) coding project folder																									
	Presentation <ul style="list-style-type: none">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 question addressed by tutor/lecturer.																									

Assessments Evaluation Section

Submission	Elements	Weight	Rate (0-5)	Total
Report	Question 1: Dataset 1	5		
	Question 1: Dataset 2	5		
	Question 2: Heap Sort	5		
	Question 2: Merge Sort	5		
	Question 3: Shortest Path	5		
	Question 3: Minimum Spanning Tree	5		
	Question 4: Dynamic Programming	5		
Presentation	Demonstration and Explanation	5		
Total (40%)				

TCP2101: ALGORITHM DESIGN AND ANALYSIS TERM 2310

Rubrics Section						
Criteria	No Attempt (0)	Very Weak (1)	Weak (2)	Average (3)	Good (4)	Excellent (5)
Part 1 dataset generation algorithm and implementations	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 and station design	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.
Merge Sort implementation,	No attempt made or implementation,	Algorithm implemented poorly	Algorithm implemented with	Algorithm implemented with	Algorithm implemented without	Algorithm implemented

TCP2101: ALGORITHM DESIGN AND ANALYSIS TERM 2310

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 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

TCP2101: ALGORITHM DESIGN AND ANALYSIS TERM 2310

				learnt concepts for real-world solutions.	real-world solutions.	concepts for real-world solutions.
Demonstration and Explanation	No attempt made or all members are not able to explain even the overall execution of the assignment or group did not attend the presentation.	Redundant explanations without highlighting main discussion points of the assignment. Implied lack of ability to relate implemented algorithm with learnt concepts.	Brief explanations with brief highlight of main discussion points of the assignment. Implied lack of ability to illustrate learnt concepts.	Moderate explanations with acceptable highlights of main discussion points of the assignment. Implied lack of ability to demonstrate learnt concepts.	Good explanations with highlights of main discussion points of the assignment. Implied ability of associating learnt concepts for real-world solutions.	Detailed explanations, creatively highlighting main discussion points of the assignment. Implied ability of integrating learnt concepts for real-world solutions.

