ASSESSMENT DOCUMENT

Module Title:

Computational Methods

Module Code: GDEV40015

Assessment No: 1

Module Leader: Craig Weightman
Room: Mellor S315

Teaching Team: Bradley Davis
Room: L105

Paul Roberts
Room: Mellor S309

James Banton
Room: Mellor S309

The Assignment

Practical Requirements

Scenario: Space Mining Game Problem

- You are in command of an inter-galactic mining vessel, which has to pick up loads from a series of 5 planets in a solar system.
- You can assume that the planets are geo-stationary with respect to each other and so maintain a fixed distance in space.
- The distances are stored in the table (adjacency matrix) below and are in millions of km.

	ALPHA	BETA	GAMMA	DELTA	EPSILON
ALPHA	-	10	15	12	20
BETA	10	-	12	25	14
GAMMA	15	12	-	16	28
DELTA	12	25	16	-	17
EPSILON	20	14	28	17	-

- The Fuel Consumption of the spaceship is based on the loading of the spaceship. The cost of the depletion of fuel is 25 intergalactic currency units per metric tonne per million kms.
- The loads picked up at each planet are as follows and assume that the spacecraft can fit all the loads inside its hold.

Alpha	**20 Tons**
Beta	**40 Tons**
Gamma	**70 Tons**
Delta	**10 Tons**
Epsilon	**30 Tons**

Perform the following tasks for the above scenario

Task 1: Brute Force

a) Draw a graph of this scenario.

b) Use a brute force approach to establish all of the possible routes. Do this in an excel spreadsheet.

c) Which is the cheapest route through this network, using this approach?

d) Explain why this approach is not optimal (No more than 500 words).

Note: If you feel comfortable with coding this solution, this can be considered to assist with the higher marks**.**

Task 2: Sorting

Below is an array of all of the edge weightings in this scenario:

10	15	12	12	25	16	20	14	28	17

a) Perform a sort algorithm on this array to sort all of the edge weightings into order. Show how this process evolves to the solution.

b) Write down pseudo-code for the algorithm you have used.

c) Describe the algorithm you have utilised, including an analysis of its complexity (No more than 500 words).

Note: If you feel comfortable with coding this solution, this can be considered to assist with the higher marks.

Note: Higher marks will be awarded for more efficient strategies used.

Task 3: Greedy Strategy

a) Using a greedy strategy of your choice, traverse the graph from any start point, visiting all destinations. Show how this progresses in diagrammatic form.

b) Write pseudo-code for your chosen solution.

c) Describe the algorithm you have used, including an analysis of its complexity (No more than 500 words).

Note: If you feel comfortable with coding this solution, this can be considered to assist with the higher marks.

Note: Higher marks will be awarded for more efficient strategies used.

Task 4: Dynamic Programming

a) Use a dynamic programming approach to find the cheapest route through the network. Do this as an excel spreadsheet.

b) Which is the cheapest route through this network, using this strategy? How does it compare with that found in the brute force scenario?

c) Perform an analysis of the efficiency of this approach, include commentary on complexity (No more than 500 words).

Task 5: NP Complete Problems

Here is a list of **problems** that are considered NP complete. For the problem corresponding to the first letter of your surname, write a short (up to 1000 words excluding references) account which:

- defines the problem and its significance
- briefly describes approaches to solving the problem.

Surname starting with:	Topic
A-C	Art Gallery Problem
D-G	Bin-Packing Problem
H - I	3-colour map colouring
J - M	Job-scheduling problem
N - T	Pancake Sorting Algorithm
U - Z	Sudoku

Workflow Requirements

You will need to do extra reading for all of these questions - use Harvard style to reference your sources. See the guide to Harvard style here. http://libguides.staffs.ac.uk/refzone/harvard

You should also consider that we do count pseudocode as an ability to apply an algorithm in a logical expression, to later be translated into any language. It is therefore possible to gain full marks with no language-based code submitted. However, there is the option in the above questions to submit programs. This will help support your submission, if there are any flaws in other parts of your submission.

We do understand that this is a first-year module, and you may be learning to program for the first time, in the same term as Computational Methods.

Submissions

Formative Guidance

As part of the week 5 tutorial session, you will be able to receive feedback on your progress. This feedback will also be available, throughout the course of the module.

Summative, Final Submission, Week 9

In week 9, you are required to submit your answers in a document, along with any supplementary material, such as spreadsheets or visual studio projects.

Assessment Criteria

Assignment Overview and Breakdown

This assignment is weighted at 100% of the module
The Marks are broken down in the following way:

Task 1: Brute Force	**15%**
Task 2: Sorting	**15%**
Task 3: Greedy Strategy	**15%**
Task 4: Dynamic Programming	**30%**
Task 5: NP Complete Problems	**25%**

Submission Deadlines

Find Submission Deadlines on the ***Assessments & Deadlines*** Tab on the module Blackboard

Submission Requirements

The table below shows the location and file formats of each piece of evidence that you are required to submit. All files should have your student number at the start of the filename and the listed suffix from the table. **Please note that all documents should be included in one overall .zip file, as detailed in the table.**

Element	Location	File Format	Suffix	Example Filename
Spreadsheets	Digital Academy Uploads	.xls / .xlsx	**_Sheet**	b235069s_Brute_Force_Sheet.xlsx
Code Solutions (If applicable)	Digital Academy Uploads	.zip	**_Code**	b235069s_Brute_Force_Code.zip
Answer document	Digital Academy Uploads	.docx	**_Answers**	b235069s_Assignment_Answers.docx

Extensions

Extensions can only be granted based on a learning support agreement (LSS). If you are entitled to an extension in accordance with you LSS, please contact the module leader to discuss extended deadlines.

Exceptional Circumstances

At Staffordshire University we understand that students may experience difficulties at some point in their studies, due to an illness or personal events. For none LSS Extensions please review the Universities Exceptional Circumstances Procedure available here

https://www.staffs.ac.uk/students/course-administration/academic-policies-and-regulations/exceptional-circumstances-procedure

Marking

Your marks will be posted to Grade Details on Blackboard when your work is graded.

Module Feedback

You will get feedback on the Digital Academy and in class sessions throughout the module.

You will also receive feedback on your grade breakdowns on Blackboard after submissions which will build upon the CRG (available on Blackboard).

Please also email staff if you wish to discuss something outside of these opportunities.

Frequently Asked Questions

How many submissions am I allowed?
You can submit as many times as you want before the deadline. However, only the latest submission submitted before the deadline will be taken into consideration.

How much help can I get on my assignment work?
As the work need to be your own (see plagiarism), you must attempt the assignment without help from your tutors. Of course, you can ask to help you understand the assignment, in general, but the work produced must be entirely student work.

I am falling behind on my studies, what should I do?
If you are struggling to keep up with the pace of the class, make sure you speak to your tutor / the module leader. Do not be scared to approach us – we are here to help.

What is classed as Plagiarism?
All work submitted must be your own. If you have utilized something that is not your own work, you will have to reference it. We can only grade you on what is yours. Not doing so can have serious ramifications.

What software Can I Use?

- Windows PC
- Visual Studio
- Word Processor
- Video Capture/Editing Software
- Blackboard for resources
- Microsoft Teams for communication and resources
- OneDrive for backing up resources
- GitHub for a more professional version control backup option

Module Learning Outcomes

The following module learning outcomes are assessed via this assessment:

Knowledge & Understanding

Demonstrate knowledge in the development of solutions to computational problems, based on algorithmic methods.

Problem Solving

Implement solutions to computational problems using either spreadsheet, or programming techniques.

Application

Code solutions according to the algorithmic techniques taught in the theory elements.

Enquiry

Apply research techniques to find and help justify understanding of mathematical principles used.

You can find additional information about the module on the module descriptor.