Semester 1, 2020

CAB301: Algorithms and Complexity

Assignment 1: Project (Applied)

Marking Criteria Sheet

Criteria /100	7 – High Distinction 85-100%	6 – Distinction 75-84%	5 – Credit 65-74%	4 – Pass 50-64%	3 – Marginal Fail 40-49%	2 – Fail 25-39%	1 – Low Fail <25%
Data structures Weighting: 10%	• You have used all the required data structures perfectly.	• You have used all the required data structures with few deficiencies.	• You have used all the required data structures with some issues.	• You have used some of the required data structures, but not all the required data structures.	• You have attempted to use required data structures, but not appropriately.	• You have not used the required data structures. You have used some inappropriate data structures.	• You have not used any data structures at all.
Algorithms	• The logic of your algorithm is correct. • Your	• The logic of your algorithm is correct. • Your	• The logic of your algorithm is correct. • Your	• The logic of your algorithm is basically correct.	• The logic of your algorithm is basically correct.	• The logic of your algorithm is not clear. • Your	• No algorithm is presented. • No algorithm

Criteria /100	7 – High Distinction 85-100%	6 – Distinction 75-84%	5 – Credit 65-74%	4 – Pass 50-64%	3 – Marginal Fail 40-49%	2 – Fail 25-39%	1 – Low Fail <25%
Weighting: 20%	algorithm is well presented using the pseudocode notations. • Your algorithm is concise and efficient. • The computational complexity of your algorithms has been analysed rigorously.	algorithm is presented using the pseudocode notations. • Your algorithm is concise and efficient. • The computational complexity of your algorithms has been analysed correctly.	algorithm is clearly presented. • Your algorithm is efficient. • The computational complexity of your algorithms has been analysed. But there are some deficiencies in your algorithm analysis.	• Your algorithm is reasonably presented. • Your algorithm is basically efficient. • You have attempted to analyse the computational complexity of your algorithm. But there are some significant issues in your algorithm analysis.	• Your algorithm is reasonably presented. • Your algorithm is not efficient. • You have not attempted to analyse the computational complexity of your algorithm.	algorithm is presented. • Your algorithm is not efficient. • You have not attempted to analyse the computational complexity of your algorithm.	analysis.
Functionality	• All the required functionality has been implemented	• All the required functionality has been implemented	• Most of the required functionality has been implemented	• Most of the required functionality has been implemented	• You have attempted to implement all the functionality.	• You have attempted to implement some of the functionality.	• You have not attempted to implement

Criteria /100	7 – High Distinction 85-100%	6 – Distinction 75-84%	5 – Credit 65-74%	4 – Pass 50-64%	3 – Marginal Fail 40-49%	2 – Fail 25-39%	1 – Low Fail <25%
Weighting: 50%	and all your implementations meet the requirements in the assignment specification. • There is no runtime problem.	and most of your implementations meet the requirements in the assignment specification. • There is no runtime problem.	and all your implementations meet the requirements in the assignment specification. • There is no runtime problem in your implementation.	and most of your implementations meet the requirements in the assignment specification. • There are some runtime problems in your implementation.	• There are some compilation problems in your implementation.	• There are some compilation problems in your implementation.	any of the functionality.
Testing Weighting: 10%	• You have provided enough screenshots for every required functionality.	• You have provided enough screenshots for most of the required functionality.	• You have provided some screenshots for most of the required functionality.	• You have provided some screenshots for some of the required functionality.	• You have provided screenshots for a small number of the functionality.	• You have provided few screenshots for a small number of the functionality.	• No screenshot is provided.
Code quality	• Your code is well structured and formatted,	• Your is well structured, allowing the	• Your code is formatted so that the logic	• Your code is mostly formatted so	• Your code is formatted so that it takes	• Your code is not formatted.	• Your code is not formatted.

Criteria /100	7 – High Distinction 85-100%	6 – Distinction 75-84%	5 – Credit 65-74%	4 – Pass 50-64%	3 – Marginal Fail 40-49%	2 – Fail 25-39%	1 – Low Fail <25%
Weighting: 10%	allowing the logic to be easily followed. • Your code is clearly and concisely described by comments that fully document the code. • Your code uses meaningful identifier names that enhance code readability by clearly explaining their purpose.	logic to be fairly easily followed. • Your code is clearly described by comments that fully document the code. • Your code uses meaningful identifier names that enhance code readability.	can be followed with minimal effort. • Your comments provide a good understanding of the code. • Your code uses meaningful identifier names.	that the logic can be followed with minimal effort. • Your comments provide a general understanding of the code. • Your code generally uses meaningful identifier names.	some effort to follow the logic. • Your comments provide little understanding of the code. • Your code uses too many unmeaningful identifier names.	• Few comments are provided. • Your code uses too many unmeaningful identifier names.	• No comment is provided. • Your code uses too many unmeaningful identifier names.