**Algorithmics Unit 4**

**School Assessed Task – “What is the best way for a tourist to spend their time?”**

**Overview**

In Unit 3, you designed an algorithmic solution for a real world problem. This was Part 1 of your School Assessed Task. In Unit 4 you will complete Parts 2 and 3.

In Part 2, you will analyse the time complexity of your solution, and establish the efficiency of your algorithm, explaining its feasibility in the real world.

In Part 3 you will use advanced algorithm designs to improve the efficiency of computation of your solution. You may add an additional constraint to the problem in order to apply the advanced algorithm design techniques that you have studied.

**Part 2**

In Part 2 you will complete a formal time complexity analysis of your algorithm and an explanation of the consequences of these results on the algorithm’s real-world application.

*You may use call graphs, recurrence relations, Master Theorem (where / if appropriate) and Big-O notation. You can evaluate the suitability of algorithms / modules based on their time complexity and recognise common features in their simple algorithms that have the same time complexity.*

You will explain the consequences of your analysis on the algorithm’s time complexity, considering discussion of input sizes and suitability to the problem’s requirements.

**Part 3**

In Part 3 you will design an improved data model and algorithm combination to solve your original problem, including:

* the selection of an efficient, coherent and fit-for-purpose solution using advanced algorithm design techniques.
* a time complexity analysis.
* a comparison to the original solution, considering tractability and time complexity.

Your improved solution should exhibit significant advantages over your original solution.

**Timeline**

Each part will be marked separately, so the timeline below must be adhered to. Penalties apply for late submission.

Part 2: Due 4pm on 28th July 2023

Part 3: Due 4pm on 25th August 2023

**Assessment Criteria**

Part 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1-2 (very low) | 3-4 (low) | 5-6 (medium) | 7-8 (high) | 9-10 (very high) |
| Determines time complexity of the original algorithmic solution. | Identifies the time complexity of some operations within the algorithm. | Identifies the time complexity of some control structures or non-constant-time sequences of operations within the algorithm.  Combines time complexity terms by applying some appropriate logic. | Identifies essential elements of the algorithm that contribute to its time complexity.  Combines time complexity terms by applying appropriate logic. | Analyses the time complexity of a sophisticated algorithm, that involves functional abstraction and/or recursion, by identifying the time complexities of a broad range of pseudocode elements and logically combining these to produce an overall result. | Analyses the time complexity of a sophisticated algorithm, that involves functional abstraction and/or recursion, by identifying the time complexities of a broad range of pseudocode elements and logically combining these to produce a tight upper-bound of the algorithm’s time complexity. |
| Explains the consequences of an algorithm’s time complexity on its real-world application. | Briefly describes how an algorithm’s running time would grow as its input size increases. | Describes how an algorithm’s running time would grow as its input size increases, based on an understanding of its time complexity. | Explains some consequences of an algorithm’s time complexity on its real-world application, based on how the algorithm’s running time would grow as its input size increases. | Explains the consequences of an algorithm’s time complexity on its real-world application, including a discussion of practical input sizes and its suitability to the problem’s requirements. | Clearly and precisely explains the consequences of an algorithm’s time complexity on its real-world application, including a thorough discussion of practical input sizes and its suitability to the problem’s requirements |

Part 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1-2 (very low) | 3-4 (low) | 5-6 (medium) | 7-8 (high) | 9-10 (very high) |
| Describes the design of an improved algorithmic solution to the real-world/applied problem. | Identifies an algorithm design approach and describes limited aspects of a design for an improved data model and algorithm combination. | Describes an algorithm design approach and specifies the overall structure of a design for an improved data model and algorithm combination. | Designs an improved data model and algorithm combination that considers advanced algorithm design approaches and provides a clear description of the solution. | Designs an improved data model and algorithm combination utilising advanced algorithm design approaches and/or sophisticated combinations or modifications of algorithms.  Clearly describes the solution, communicating the algorithm in pseudocode. | Designs an improved data model and algorithm combination utilising advanced algorithm design approaches and/or innovative combinations or modifications of algorithms.  Succinctly and precisely describes the solution, communicating the algorithm in pseudocode. |
| Quality of the improved solution. | The improved solution exhibits limited advantages over the initial solution with regard to its efficiency, coherence and fitness for purpose. | The improved solution exhibits some advantages over the initial solution with regard to its efficiency, coherence and fitness for purpose. | The improved solution exhibits a range of advantages over the initial solution with regard to its efficiency, coherence and fitness for purpose. | The improved solution exhibits many advantages over the initial solution with regard to its efficiency, coherence and fitness for purpose. | The improved solution exhibits considerable advantages over the initial solution with regard to its efficiency, coherence and fitness for purpose. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Compares algorithmic solutions in terms of their coherence and fitness for purpose. | Identifies some points of comparison between the algorithmic solutions to the real-world/applied problem. | Outlines some points of comparison between the algorithmic solutions based on their design features or fitness as solutions to the real-world/applied problem. | Describes some points of comparison between the algorithmic solutions based on their design features and fitness as solutions to the real-world/applied problem. | Compares the algorithmic solutions based on their design features, coherence and fitness as solutions to the real-world/applied problem, including the identification of their similarities and differences. | Comprehensively compares the algorithmic solutions based on their design features, coherence and fitness as solutions to the real-world/applied problem, including the thorough identification of their similarities and differences. |
| Compares whether the solutions will render the problem tractable and the real-world implications of this.  Compares the relative efficiency of the solutions with regard to the constraints of the real-world/applied problem context. | Identifies some points of comparison between the algorithmic solutions in relation to whether they would render the problem tractable.  Identifies some points of comparison between the algorithmic solutions in terms of their relative efficiency based on their time complexities. | Outlines some points of comparison between the algorithmic solutions as to whether they would render the problem tractable.  Outlines some comparison points between the algorithmic solutions in terms of their relative efficiency based on their time complexities and considering the constraints of the realworld problem. | Describes some points of comparison between the algorithmic solutions in relation to whether they would render the problem tractable.  Describes some advantages disadvantages of the algorithmic solutions in terms of their relative efficiency based on their time complexities and considering constraints of the real-world problem. | A comparison of whether the solutions would render the problem tractable and a discussion of the relevant implications of this to their potential application to the problem.  A well-developed and considered comparison of the relative efficiency of the solutions, with regard to their time complexities and the constraints of the real-world/applied problem. | A comprehensive comparison of whether the solutions would render the problem tractable and a thorough discussion of the relevant implications of this to their potential application to the problem.  A precise and comprehensive comparison of the relative efficiency of the solutions, with regard to their time complexities and the constraints of the real-world/applied problem. |