**Algorithmics Unit 3**

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

**Overview**

1. Create a model of a public transport system in a particular city or region.
2. Design an algorithm to find the best way for a tourist to spend their time.
3. Evaluate your model/algorithm.

A basic SAT will only consider the distances/times between nodes. A more advanced SAT will include other salient features, such as the popularity of certain locations and the time taken to embark/disembark at various locations.

Further information/guidance will be given in online lessons and on the news forum on VSV online.

**Model**

* Clearly specify the problem, including discussion of the relevant real life information you intend to model.
* 15-25 nodes
* Uses combinations of ADTs

**Algorithm**

* Is chosen from a range of potentially appropriate algorithms
* signatures provided for key operations
* Combines approaches/algorithms as appropriate to solve your stated problem

**Evaluation**

* Justifies your choice of model, its coherence and fitness for purpose.

**Authentication**

* This is an individual project. You can discuss it with others, but the work must be your own.
* Students must be able to explain their work if called upon.
* You must acknowledge any external resources you draw upon.

**Timeline**

By 28th April: you have chosen a real life situation, specified the problem you intend to solve, and how you will model it with ADTs. You can submit this work earlier if you would like to receive feedback earlier.

By 12th May: you have modelled your situation using Edgy/Python and made progress on your algorithm design.

By 2nd June: you have submitted your final project.

You will receive feedback at each stage. Your final report will be assessed against all criteria, so you can improve your score from previous submissions. Penalties apply to submissions received after 2nd June.

**Assessment Criteria**

Criterion 1: specify a problem and model its key features. Precisely specify the algorithm problem you intend to solve.

Criterion 2: design an algorithm to solve your problem, considering multiple approaches and combinations of algorithm patterns. 600-800 words.

Criterion 3: communicate your algorithm using pseudocode.

Criterion 4: evaluate your model, considering the suitability of your algorithm, how it integrates with your model and whether or not it solves your original problem. (Formal analysis will come later in Unit 4). 300-500 words.

You will receive a mark out of ten for each criterion, based on the following rubric.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 1-2 (very low) | 3-4 (low) | 5-6 (medium) | 7-8 (high) | 9-10 (very high) |
| 1 | Specifies an algorithmic problem. | Identifies some algorithmic aspects of the real world problem. | Outlines/explains some algorithmic aspects of the real world problem. | Formulates an algorithmic problem from the real world problem. | Explains how the algorithmic problem is formulated from the real world problem. | Provides a clear and precise specification of the algorithmic problem, formulated from the real world problem |
| 1 | Explains the salient features of the real world problem. | Lists arbitrary features of the real world problem. | Identifies some relevant features of the real world problem, giving reasons. | Describes salient features of the real world problem, giving reasons. | Chooses a suitable set of features to model based on consideration of the features of the real world model. | Chooses a **comprehensive** set of features to model based on consideration of the features of the real world model |
| 1 | Models the problem using ADTs. | Makes a limited attempt, ADTs may not be suitable. | Models some features using suitable ADTs. | Models selected features using a combination of suitable ADTs. Outlines how the problem maps to this model. | Models selected features using a **coherent** combination of suitable ADTs. Describes how the problem maps to this model. **Some signatures** are provided. | Models selected features using a **coherent and fit-for-purpose** combination of suitable ADTs. Describes how the problem maps to this model. **All signatures** for key operations are provided. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 1-2 (very low) | 3-4 (low) | 5-6 (medium) | 7-8 (high) | 9-10 (very high) |
| 2 | Considers suitable algorithmic approaches to the problem | Identifies an algorithm with some relevant to the problem | Outlines a few algorithm design approaches that could form the basis of a solution. | Considers relevant characteristics of several algorithm design approaches. | Compares the suitability of some algorithm design approaches to determine an appropriate approach. | **Thoroughly** compares the suitability of algorithm design approaches to determine the best approach. |
| 2 | Describes the design of an algorithmic solution to the real world problem | Identifies some aspects of an algorithm to solve a real world problem. | Outlines a simple solution to the problem. | Describes a non-trivial algorithm that some some aspects of the specified problem. | **Describes** an algorithm that solves the specified problem using some combination of algorithms or algorithm design patterns. | **Clearly explains** an algorithm that solves the specified problem using some combination of algorithms or algorithm design patterns. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 1-2 (very low) | 3-4 (low) | 5-6 (medium) | 7-8 (high) | 9-10 (very high) |
| 3 | Communicates the algorithmic solution in pseudocode | Limited elements are expressed in pseudocode. Some correct initialisation of variables and data structures. | Elements of the structure are expressed in pseudocode. Correct use of simple iteration and conditional control structures where appropriate. | Algorithm is expressed in pseudocode so that the structure of the design is clear. Includes nested iteration and recursion where appropriate. | Algorithm is expressed in pseudocode that mostly reflects the solution design. Errors are minor and do not affect the overall structure. Some attempt to use functional abstractions. | Algorithm is expressed in pseudocode correctly and precisely. A modular approach is used including ADTs and functional abstractions. |
| 4 | Justifies the solution to the real world problem. | Identifies relevant reasons in support of a solution. | Outlines the rationale for a solution based on limited consideration of pros and cons. | Justifies choice of solution based on comparative advantages over other approaches. | Justifies choice of solution based on arguments relating to its suitability, coherence or fitness for purpose. | Justifies choice of solution by clearly demonstrating its suitability, coherence or fitness for purpose. |