

# Feedback Form, ECM2423 Coursework

Ayah helal

## Question 1

### Question 1.1

Describe how you would frame the 8-puzzle problem as a search problem.

**Total Marks:** 5

**Awarded Mark:** 1

#### Comments:

1. The initial state of the problem is not mentioned.
2. The goal state of the problem is not mentioned.
3. All allowed moves of the problem is not mentioned.
4. Good general understanding but looking for specific initial / goal states and where / when certain moves are possible (depending on the position of the 0 node).

## Question 1.2

### Question 1.2.1

In this question, you should first briefly outline the A\* algorithm.

**Total Marks:** 5

**Awarded Mark:** 3

#### Comments:

1. Missing to mention that it is complete search.
2. Missing to mention that it keeps all nodes in memory

### Question 1.2.2

Describe two admissible heuristic functions for the 8-puzzle problem and explain why they are admissible. Make sure you also explain why you chose these two heuristic functions in particular amongst all the possible ones.

**Total Marks:** 5  
**Awarded Mark:** 5

**Comments:**

#### **Question 1.2.3**

Then, you should implement two versions of the A\* algorithm in Python to solve the 8-puzzle using the two heuristic functions you have chosen. You can either implement the two versions in the same Python script, letting the user select which one to use before running the code, or you can have two different scripts if you prefer. To test that it works, you can use the start and goal state of Figure 1 (however, note that this may take a few minutes to run depending on your computer, implementation and choice of heuristic functions), or you can specify your own initial and goal state. If you specify your own initial and goal state, select states which are at least five moves apart from each other and write these states in your report. You will not be penalised if you do not use the start and goal configurations of Figure 1, but you will not receive full marks if the configurations you select are too easy.

**Total Marks:** 10  
**Awarded Mark:** 10

**Comments:**

#### **Question 1.2.4**

Briefly discuss and compare the results given by A\* when using the two different heuristic functions in question 1.2.

**Total Marks:** 5  
**Awarded Mark:** 5

**Comments:**

#### **Question 1.3**

Write a general version of the A\* algorithm (using either of the two heuristic functions described above) to solve a generic version of the 8-puzzle where the user can input any start and goal state.

**Total Marks:** 5  
**Awarded Mark:** 5

**Comments:**

## Question 2

### Question 2.1

#### Question 2.1.1

In first problem design your evolutionary algorithm addressing the following points in your design process

**Total Marks:** 10

**Awarded Mark:** 10

**Comments:**

#### Question 2.1.2

Then, you should implement the evolutionary algorithm in Python to solve the Sudoku problem. You will need to run experiments for the three Sudoku grids provided on the ELE page, for population sizes 10, 100, 1000, 10000. Each experiment (i.e., a specific combination of grid and population size) needs to be ran 5 times (each one with a different random seed) and average performance across runs considered. In total these amount to  $3 \times 4 \times 5 = 60$  runs.

**Total Marks:** 15

**Awarded Mark:** 15

**Comments:**

### Question 2.2

Analysis the Sudoko problem using Evolutionary algorithm.

**Total Marks:** 15

**Awarded Mark:** 15

**Comments:**

## Code Presentation

**Total Marks:** 5

**Awarded Mark:** 5

**Comments:**

## Report Presentation

**Total Marks:** 5

**Awarded Mark:** 5

**Comments:**

## **Extra Experiments**

**Total Marks:** 15

**Awarded Mark:** 15

**Comments:**

## **Total**

**Awarded Mark:** 94

**Comments:**