Reflective Report of Implementation Approach

The java program has really challenged and pushed me to do my own research and revisit previous lectures in order to successfully implement the program. My gratitude goes to Dr Ayorkor Korsah for taking me through breadth-first search and uniform search cost search in the Introduction to AI class. So basically, I used a uniform cost search to design and implement my work. In the process of writing the code, I utilized a priority queue, a node class, the comparator class, assert keyword, buffered Reader and Writer, File Writer, regex and error handling just to point out the main aspect of the work.

The initial idea was to design an optimal solution using the flight distance. Therefore, I thought it best to employ a uniform cost search. I later switched to using the number of flights for the optimal solution. I designed a public node class which has instance variables: airport current city, parent airport city and trip cost. A priority queue was used to store the nodes to easily pick the node with the lowest trip cost for processing. To make comparison easy using the trip cost as the yardstick, I used the comparator class to create an object, which specified the trip cost and passed the comparator object as an argument in the priority queue.

In all, I had six classes namely: Node, fileWriter, fileReader, ProblemSpace, Map and computation. The fileReader returns an ArrayList of the information from the location (and file) passed into it as an argument. Likewise, the fileWriter creates a file in the current location and outputs the flight information in the created file. The ProblemSpace defines the starting city, starting country, destination city and destination country as static variables. The Map returns a Hash table for each airport city in the route file and an ArrayList of all the airport cities (value) the respective airport cities (key) can move to.

While implementing the program, I learnt how to use regex (regular expression) to avoid the ‘,’ problem and also used assert to check if a code I wrote is reachable or not.