**CSC-462 ARTIFICIAL INTELLIGENCE**

**LAB 06**

**Graph Theory and Path Searches in Python**

**Manaal Waseem**

**FA18-BCE-074**

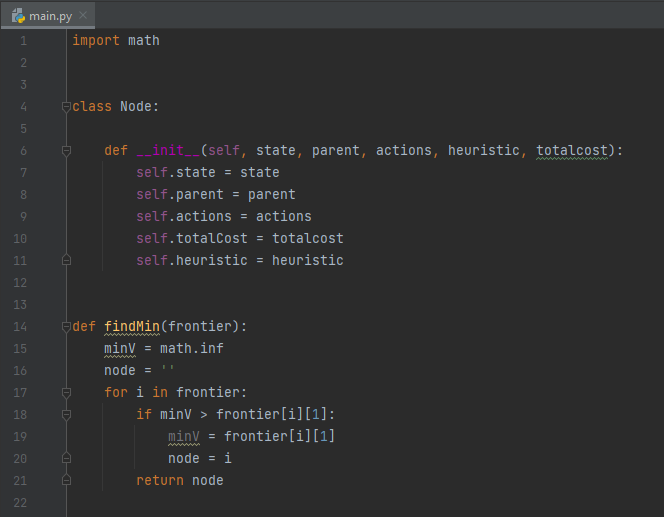
**BCE-7B**

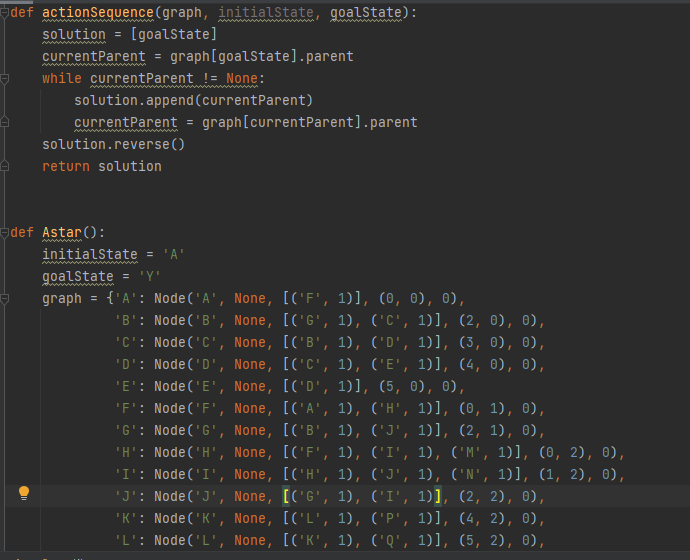
**Activities:**

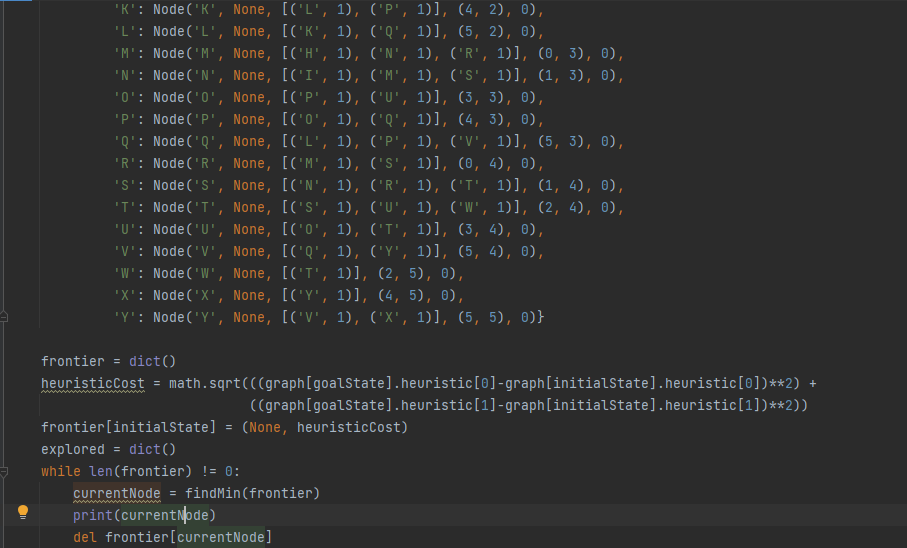
**Activity 1:**

Consider a toy problem that can be represented as a following graph. How would you represent this graph in python?

Change initial state to D and set goal state as C. What will be resulting path of A\* search?







Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

In this lab task, we were to implement A\* on graphs or trees to find the goal state. The algorithm works in a way that it checks the heuristic cost and the total cost to reach the child node. It selects the child node who has the least sum of path cost and heuristic cost. In this way it chooses an optimal solution to reach the goal state with minimum total cost.

**Activity 2:**

Consider a toy problem that can be represented as a following graph. How would you represent this graph in python?

Change initial state to D and set goal state as C. What will be resulting path of hill climbing search?

A picture containing text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

In this lab task, we were to implement hill climbing search on graphs or trees to find the goal state. The algorithm works in a way that it checks the lowest value and according to the lowest peak would it keeps on moving forward from node to node in order to find the lowest peak in the graph. In this code we have set the goal state to Y, but the solution doesn’t go to Y because before Y there is J and the algorithm thinks J as the lowest peak because after J come G which had a higher peak than the J node.

**Critical Analysis:**

This lab was to get an introduction on python language. In this lab we implemented A\* and hill climbing search to reach our goal states.