**Project 2 Artificial Intelligence :CSCE 5210 – Fall 2021**

**Distributed: Tuesday, September 28**

**Due: Tuesday October 19**

Q1) We need a queue and tree data structure to support the minimization of movement between divisions.

**Queue** in linear data structure to save the current division node that we visited, so we can record the path and find the shortest path from this recorded queue.

**Tree** in non-linear data structure to create the map of this binary tree which is a bi-directional movement.

Q2) Outline the procedure :

The agent starts searching the items(k) by going into the division binary tree which consists of fifteen division nodes. Additionally, these division nodes have the path’s cost randomly from 10-40. Inside the division nodes, the agent needs to search for the item in the subtree (shelf binary trees), and each tree has 63 nodes. The path’s cost of these nodes is one. Here are the attributes in project two.

1. Creating the customer orders’ data set for this project
   1. Creating **the division number** in the range 1 to 15.
   2. Creating **the shelves locations** in the range 1 to 63 which are generated that represent the shelf numbers where the items are located.
   3. **The number of items** which is represented by **k**, and every customer can order 3 items per order, top.
   4. Creating **m** which is represented as **the node (the shelves)** in the division, and the lowest level of each node has two children numbered 2m and 2m+1 respectively.
   5. **Heuristic search** - Creating **the cost of moving** inside the shelves and divisions which is a constant step cost of 1. This cost will add up whenever the robot moves from any parent node to any one of its child nodes.

**Note :** Every item will be distributed randomly in the shelves and divisions.

1. Creating the queue to store the current division node that we visited.
2. Creating the tree to store the value on above
   1. Division binary tree [Main tree]
      1. Parent node
         1. two children nodes(left and right) per one parent node
      2. Level of the node : 4
      3. Path cost between each node : 10, 20,30, 40
      4. Label node with the number : 1-15
      5. Current\_node : Current position of the robot
      6. Tail node [the last node] : 8 nodes
   2. Shelf binary tree [Sub Tree]
      1. Parent node
         1. two children nodes(left and right) per one parent node
      2. Level of the node : 6
      3. Path cost between each node : 1
      4. Label node with the number : 1-63
      5. Current position of the robot
      6. Tail node[the last node] : 32 nodes

Q3) Implement Q2 above by writing a Python program that populates the data structure you proposed in

Please see the code

Q4 Implement Iterative Deepening Search (IDS) to locate shelf locations for servicing customer orders.

Please see the code

Q5) Test your program for this boundary case. Generate an order for division 6 and trace its path from division 1 to division 6 and then onwards to the only item, item 33, that was ordered. Check that its path length is what you expected. Print out both the path and its length.

Please see the code

Q6) Run your program for a certain number N of customer orders (say 100) and answer the following questions:

1.What is the average path length travelled across the N orders? Take care to include in your path length both the paths travelled to get to the division that contains the order and the movement required within the division to get to the items required.

2.Print out the length of the shortest and longest paths across the N orders that you generated.

The shortest division path of length 0: []

The longest division path of length 15: [1, 2, 4, 8, 9, 5, 10, 11, 3, 6, 12, 13, 7, 14, 15]

The shortest shelf path of length 0: []

The longest shelf path of length 57: [7, 14, 28, 56, 57, 29, 58, 59, 15, 30, 60, 61, 31, 62, 63, 3, 6, 12, 24, 48, 49, 25, 50, 51, 13, 26, 52, 53, 27, 54, 55, 1, 2, 4, 8, 16, 32, 33, 17, 34, 35, 9, 18, 36, 37, 19, 38, 39, 5, 10, 20, 40, 41, 21, 42, 43, 11]