Subject: Advanced Intelligence Systems
Assignment: Road Warrior – Part 01 (03/01/2021)

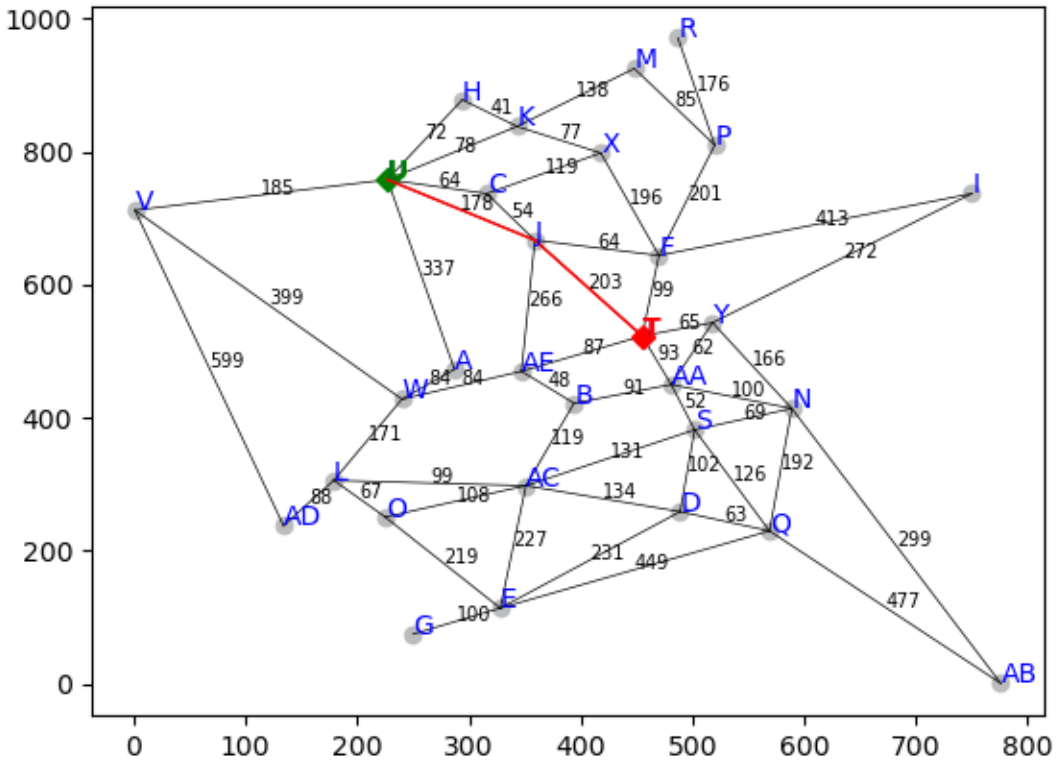
By
Saurabh Jawahar Kakade
Sk2354@nau.edu

1. Solution: Printing map from start node U to goal node T in 30node1.png
   Related Code:

```
# (b) Show your program loading in the 30-node sample file.
s = Searcher("30node.txt")

# (c) Show you program setting start node=U and end node=T.
s.setStartGoal('U','T')

# myViz should be a DRDViz instance -> save map to file on disk.
s.myViz.save("30node1.png")
```



2. Output for:
    a. Path from U to T (BFS algorithm)
    b. First open node = s.open() code command is used
    c. Successor of only open node = s.successor() code command is used

```
PS N:\Spring 2021\Subjects\AI\Assignments\Asssignment 02> python testfile2.py
['U', 'J', 'T']
U
['A', 'C', 'H', 'J', 'K', 'V']
PS N:\Spring 2021\Subjects\AI\Assignments\Asssignment 02>
```

3. BFS Code:

```python
def bfs_shortest_path(graph, start, goal):
    # keep track of explored nodes
    explored = []
    # keep track of all the paths to be checked
    queue = [[start]]

    # print(graph)

    # return path if start is goal
    if start == goal:
        return "That was easy! Start = goal"

    # keeps looping until all possible paths have been checked
    while queue:
        # pop the first path from the queue
        path = queue.pop(0)
        # get the last node from the path
        node = path[-1]
        if node not in explored:
            neighbours = graph[node]
            # go through all neighbour nodes, construct a new path and push it into the queue
            for neighbour in neighbours:
                new_path = list(path)
                new_path.append(neighbour)
                queue.append(new_path)
                # return path if neighbour is goal
                if neighbour == goal:
                    return new_path

            # mark node as explored
            explored.append(node)


    return "Error"
```