Ai Applications & Ethics (TC-7)

LAB 3 - A* Algorithm

Utkarsh Bhangale

20200802124

Code -

```
def aStarAlgorithm(start_node, stop_node):
    open_set = set(start_node)
    close_set = set()
    g = {} #store distance from starting node
    parents = {}

    g[start_node] = 0
    parents[start_node] = start_node

    while len(open_set) > 0:
        n = None

        for v in open_set:
            if n == None or g[v] + heuristic(v) < g[n] + heuristic(n):
                n = v

        if n == stop_node or Graph_nodes[n]==None:
            pass
        else:
            for (m,weight) in get_neighbours(n):

                if m not in open_set and m not in close_set:
                    open_set.add(m)
                    parents[m] = n
                    g[m] =g[n] + weight

                else:
                    if g[m] > g[n] + weight:
                        #update g(n)
                        g[m] = g[n] + weight
                        #change parent of m to n
                        parents[m] = n

                        if m is close_set:
                            close_set.remove(m)
                            open_set.add(m)
```

```
        if n == None:
            print ('Path does not exist!')
            return None
        if n == stop_node:
            path = [ ]
            while parents [n] != n:
                path.append(n)
                n = parents[n]
            path.append(start_node)
            path.reverse()

            print('Path found : {}'.format(path))
            return path

        open_set.remove(n)
        close_set.add(n)

    print('Path Does not exist!')
    return None

def get_neighbours(v):
    if v in Graph_nodes:
        return Graph_nodes[v]
    else:
        return None

def heuristic(n):
    H_dist = {
        's' : 14,
        'b' : 12,
        'c' : 4,
        'd' : 2,
        'e' : 6,
        'f' : 11,
        'g' : 0
    }

    return H_dist[n]
Graph_nodes = {

    's' : [('b',4), ('c',3)],
    'b' : [('f',5), ('e',12)],
    'g' : None ,
    'c' : [('e',10), ('d',7)],
    'd' : [('e',2)],
    'e' : [('g',5)],
    'f' : [('g',10)]
```

```
}

aStarAlgorithm('s', 'g')
```

Output: -

```
PS C:\Users\utkar\Desktop\College\AI APP AND ETH\lab2> & C:/Use
ege/AI APP AND ETH/lab2/Astar.py"
Path found : ['s', 'c', 'd', 'e', 'g']
PS C:\Users\utkar\Desktop\College\AI APP AND ETH\lab2>
```

Conclusion: -

Successfully implemented A* Heuristic search algorithm.