Lab Assignment-2

Contents

```
graph = { '5' : ['3','7'], '3' : ['2',
'4'],
'7' : ['8'], '2' : [], '4' : ['8'], '8' : [] }
visited = [] # List for visited
nodes.
 queue = [] #Initialize a queue
def  bfs(visited, graph, node):
#function for BFS
visited.append(node)
queue.append(node)
while queue:
  #Creating loop to visit each
  node
m = queue.pop(0)
print (m, end = " ")
for neighbour in graph[m]:
if  neighbour  not  in
visited:
visited.append(neighbo
ur)
queue.append(neighbo
ur)
# Driver Code
print("Following  is  the
Breadth-  First Search")
bfs(visited,  graph,  '5')  #
function  calling
```

A* Algorithm

```
def h(self, n):
    H = {
        'A': 1,
        'B': 1,
        'C': 1,
        'D': 1
    }

    return H[n
```

```
open_lst = set([start])
closed_lst = set([])


pool = {}
pool[start] = 0


par = {}
par[start] = start
```

```
if m not in open_lst and m not in
closed_lst:

open_lst.add(m)

// Add condition to check the
node with minimum f(n)

open_lst.remove(n)

closed_lst.add(n)
```

15 Puzzle problem using A* & BFS Search Algorithm

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Initial State

1	5	2	3
4		6	7
8	9	10	11
12	13	14	15

Goal State

Traveling Salesman Problem- Distance Matrix of 13 Cities- Find the optimal path from city 1 and calculate the distance.

```
data['distance_matrix'] = [
    [0, 2451, 713, 1018, 1631, 1374, 2408, 213, 2571, 875, 1420, 2145, 1972],
    [2451, 0, 1745, 1524, 831, 1240, 959, 2596, 403, 1589, 1374, 357, 579],
    [713, 1745, 0, 355, 920, 803, 1737, 851, 1858, 262, 940, 1453, 1260],
    [1018, 1524, 355, 0, 700, 862, 1395, 1123, 1584, 466, 1056, 1280, 987],
    [1631, 831, 920, 700, 0, 663, 1021, 1769, 949, 796, 879, 586, 371],
    [1374, 1240, 803, 862, 663, 0, 1681, 1551, 1765, 547, 225, 887, 999],
    [2408, 959, 1737, 1395, 1021, 1681, 0, 2493, 678, 1724, 1891, 1114, 701],
    [213, 2596, 851, 1123, 1769, 1551, 2493, 0, 2699, 1038, 1605, 2300, 2099],
    [2571, 403, 1858, 1584, 949, 1765, 678, 2699, 0, 1744, 1645, 653, 600],
    [875, 1589, 262, 466, 796, 547, 1724, 1038, 1744, 0, 679, 1272, 1162],
    [1420, 1374, 940, 1056, 879, 225, 1891, 1605, 1645, 679, 0, 1017, 1200],
    [2145, 357, 1453, 1280, 586, 887, 1114, 2300, 653, 1272, 1017, 0, 504],
    [1972, 579, 1260, 987, 371, 999, 701, 2099, 600, 1162, 1200, 504, 0],
]
```