

AI LAB EXP – 5a

BFS ALGORITHM FOR REAL WORLD PROBLEMS

Tejas Ashok

RA1911030010090

AIM

To implement Best First Algorithm using python.

ALGORITHM

- Define a list, OPEN, consisting solely of a single node, the start node, s.
- IF the list is empty, return failure.
- Remove from the list the node n with the best score (the node where f is the minimum), and move it to a list, CLOSED.
- Expand node n.
- IF any successor to n is the goal node, return success and the solution (by tracing the path from the goal node to s).
- FOR each successor node:
 1. Apply the evaluation function, f, to the node.
 2. IF the node has not been in either list, add it to OPEN.
- Looping structure by sending the algorithm back to the second step.

CODE

```
from queue import PriorityQueue
v = 14
graph = [[] for i in range(v)]

def best_first_search(source, target, n):
    visited = [0] * n
    visited[0] = True
    pq = PriorityQueue()
    pq.put((0, source))
```

```

while pq.empty() == False:
    u = pq.get()[1]
    print(u, end=" ")
    if u == target:
        break
    for v, c in graph[u]:
        if visited[v] == False:
            visited[v] = True
            pq.put((c, v))
    print()

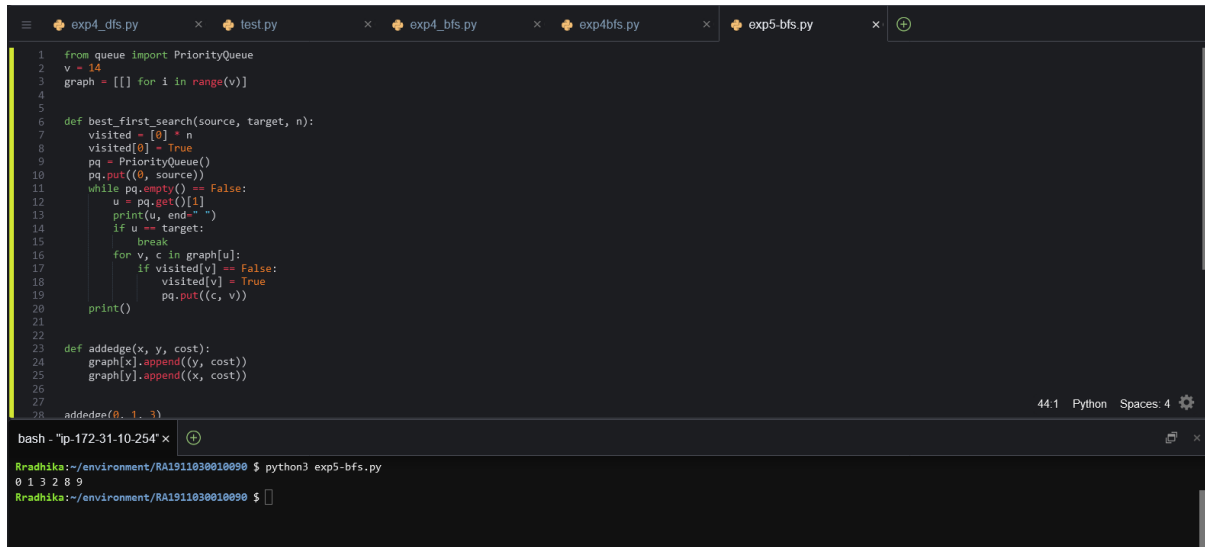
def addedge(x, y, cost):
    graph[x].append((y, cost))
    graph[y].append((x, cost))

adddedge(0, 1, 3)
adddedge(0, 2, 6)
adddedge(0, 3, 5)
adddedge(1, 4, 9)
adddedge(1, 5, 8)
adddedge(2, 6, 12)
adddedge(2, 7, 14)
adddedge(3, 8, 7)
adddedge(8, 9, 5)
adddedge(8, 10, 6)
adddedge(9, 11, 1)
adddedge(9, 12, 10)
adddedge(9, 13, 2)
source = 0
target = 9

```

```
best_first_search(source, target, v)
```

OUTPUT



```
1 from queue import PriorityQueue
2 v = 14
3 graph = [[] for i in range(v)]
4
5
6 def best_first_search(source, target, n):
7     visited = [0] * n
8     visited[0] = True
9     pq = PriorityQueue()
10    pq.put((0, source))
11    while pq.empty() == False:
12        u = pq.get()[1]
13        print(u, end=" ")
14        if u == target:
15            break
16        for v, c in graph[u]:
17            if visited[v] == False:
18                visited[v] = True
19                pq.put((c, v))
20    print()
21
22
23 def addedge(x, y, cost):
24     graph[x].append((y, cost))
25     graph[y].append((x, cost))
26
27
28 addedge(0, 1, 3)
```

```
bash - "ip-172-31-10-254" x
Radhika:~/environment/RA1911030010090 $ python3 exp5-bfs.py
0 1 3 2 8 9
Radhika:~/environment/RA1911030010090 $
```

RESULT

Best first search algorithm was successfully executed in python.