Name: Richard Nadar
Reg no: RA1911030010109
Date: 15-2-22

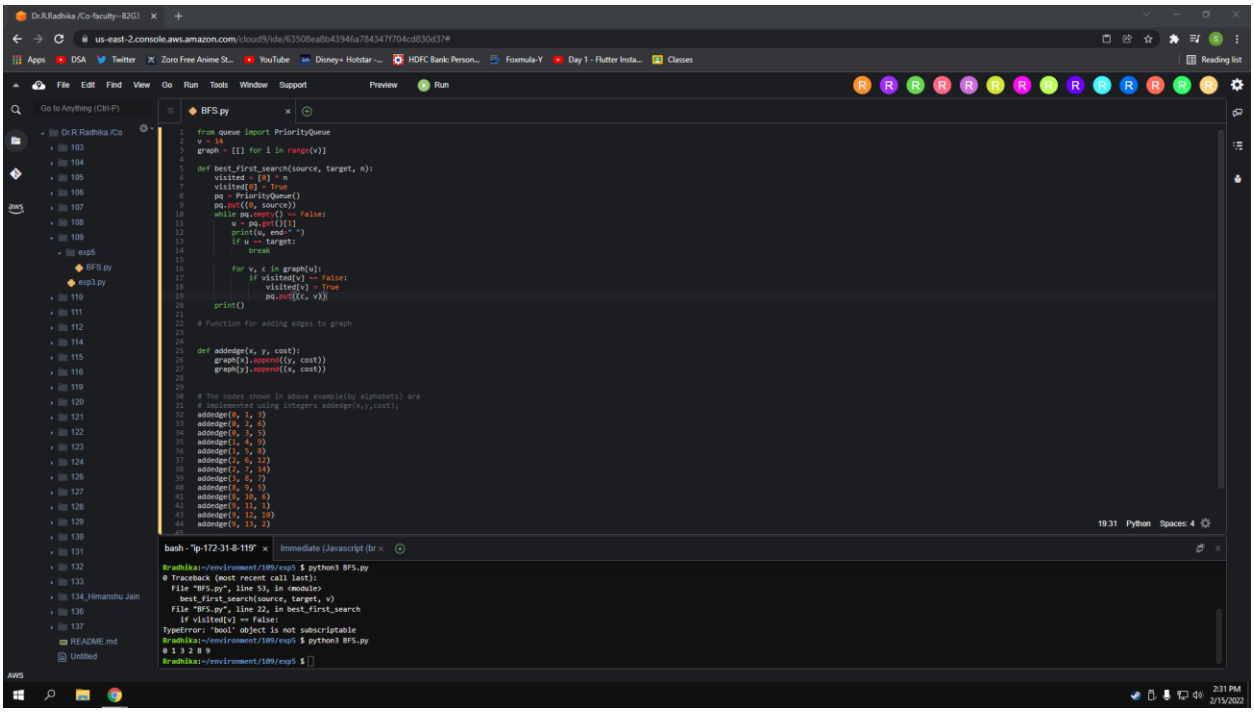
EXP 5: Implementation of Best First Search

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()
```

```
# Function for adding edges to graph
def addedge(x, y, cost):
    graph[x].append((y, cost))
    graph[y].append((x, cost))

# The nodes shown in above example(by alphabets) are
# implemented using integers addedge(x,y,cost);
addedge(0, 1, 3)
addedge(0, 2, 6)
addedge(0, 3, 5)
addedge(1, 4, 9)
addedge(1, 5, 8)
addedge(2, 6, 12)
addedge(2, 7, 14)
addedge(3, 8, 7)
addedge(8, 9, 5)
addedge(8, 10, 6)
addedge(9, 11, 1)
addedge(9, 12, 10)
addedge(9, 13, 2)

source = 0
target = 9
best_first_search(source, target, v)
```

OUTPUT:


BFS.py
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()
# Function for adding edges to graph
def addedge(x, y, cost):
graph[x].append((y, cost))
graph[y].append((x, cost))
# The nodes shown in above example(by alphabets) are
# implemented using integers addedge(x,y,cost);
addedge(0, 1, 3)
addedge(0, 2, 6)
addedge(0, 3, 5)
addedge(1, 4, 9)
addedge(1, 5, 8)
addedge(2, 6, 12)
addedge(2, 7, 14)
addedge(3, 8, 7)
addedge(8, 9, 5)
addedge(8, 10, 6)
addedge(9, 11, 1)
addedge(9, 12, 10)
addedge(9, 13, 2)
bash - "ip-172-31-8-119"
Immediate (Javascript (br
@radhika:~/environment/109/exp5 $ python3 BFS.py
Traceback (most recent call last):
File "BFS.py", line 53, in <module>
best_first_search(source, target, v)
File "BFS.py", line 22, in best_first_search
if visited[u] == False:
TypeError: 'bool' object is not subscriptable
@radhika:~/environment/109/exp5 $ python3 BFS.py
0 1 3 2 8 9
@radhika:~/environment/109/exp5 $