**AI LAB EXP – 5**

**IMPLEMENTATION OF BEST FIRST SEARCH**

***Submitted by***

**Vinoth S**

**RA1911030010103**

**GitHub Link: https://github.com/vk1308**

**AIM:**

To implement the BFS algorithm using python

**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))

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)

**Time Complexity and Space Complexity:**

Best First Search is simply Breadth First Search (BFS) but with the nodes re-ordered by their heuristic value. In the worst case, the time and space complexity for Best First Search are the same as that of BFS: O(b^(d+1)) for Time and O(b^d) for Space.

**Output:**



