**Name : Om Chandrakant Mahajan Date :-21/02/2024**

**Batch :- B3 Roll no:- 88**

**Practical no:- implementation of best first search**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

from queue import PriorityQueue

def best\_first\_search(graph, start, goal, heuristic):

queue = PriorityQueue()

queue.put((heuristic[start], start))

visited = set()

print('The path traversed is:')

while not queue.empty():

\_, current = queue.get()

print(current, " ")

if current == goal:

return True

visited.add(current)

for neighbor in graph[current]:

if neighbor not in visited:

priority = heuristic[neighbor]

queue.put((priority, neighbor))

return False

graph = {

'X': ['y', 'z'],

'Y': ['w', 'u'],

'Z': ['t'],

'w': [],

'U': ['v’],

'T': [],

'V': []

}

heuristic = {

'X': 10,

'Y': 5,

'Z': 8,

'W': 4,

'U': 3,

'T': 2,

'V': 6

}

start\_node = 'X'

goal\_node = 'V'

result = best\_first\_search(graph, start\_node, goal\_node, heuristic)

if result:

print("Goal reached!")

else:

print("No solution found.")

output:-

The path traversed is:

X

Y

U

W

V

Goal reached!