Academic Year: 2024-2025
Semester: V
Class / Branch: TE/CSE-DS
Subject: **Artificial Intelligence Lab**
Name of Instructor: Prof. Sarala Mary

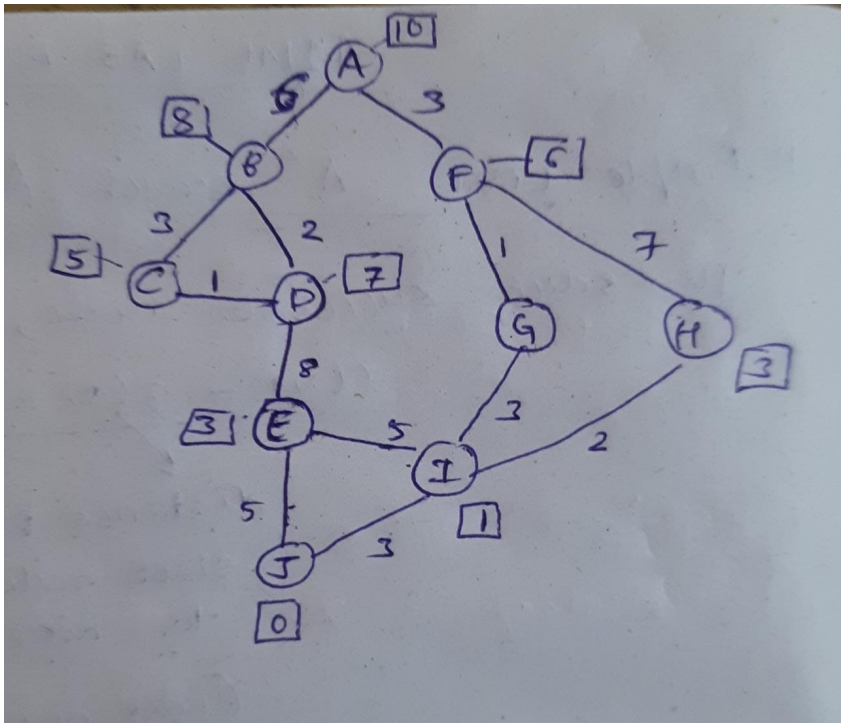
Name of Student:Gauri Iyer
Student ID: 22107030
Date Of Performance: 19/8/24
Date Of Submission: 19/8/24

Experiment No.04

Aim:- To implement A* algorithm using python for the given graph

Program:

1. Write a program to implement A* algorithm using python for the given graph.



Code:

```
Graph_nodes={
'A':[('B',6),('F',3)],
'B':[('C',3),('D',2)],
'C':[('D',1),('E',5)],
'D':[('C',1),('E',8)],
'E':[('I',5),('J',5)],
'F':[('G',1),('H',7)],
'G':[('I',3)],
'H':[('I',3)],
'I':[('E',5),('J',3)]
}

def get_neighbours(v):
if v in Graph_nodes:
return Graph_nodes[v]
else:
return None

def h(n):
H_dist={
'A':10,
'B':8,
'C':5,
'D':7,
'E':3,
'F':6,
'G':5,
'H':3,
'I':1,
'J':0
}
return H_dist[n]

def aStarAlgo(start_node,stop_node):
open_set=set(start_node)
closed_set=set()
```

```
g={}
parents={}
g[start_node]=0
parents[start_node]=start_node

while len(open_set)>0:
n=None

for v in open_set:
if n==None or g[v]+h(v)<g[n]+h(n):
n=v
if n==stop_node or Graph_nodes[n]==None:
pass
else:
for (m,weight) in get_neighbours(n):
if m not in open_set and m not in closed_set:
open_set.add(m)
parents[m]=n
g[m]=g[n]+weight
else:
if g[m]>g[n]+weight:
g[m]=g[n]+weight
parents[m]=n
if m in closed_set:
closed_set.remove(m)
open_set.add(m)
if n==None:
print('Path doesnt exist!')
return None
if n==stop_node:
path=[]
while parents[n]!=n:
path.append(n)
n=parents[n]
path.append(start_node)
path.reverse()
print('Path found: {}'.format(path))
return path
```

```
open_set.remove(n)
closed_set.add(n)
print('Path  doesnt  exist!')
return  None
aStarAlgo('A','J')
```

Output:

```
Path found: ['A', 'F', 'G', 'I', 'J']
['A', 'F', 'G', 'I', 'J']
```