***Artificial Intelligence***

***CSL 411***

***Lab Journal 6***

****

**Student Name:Rida Sajjad**

**Enrolment No.01-134152-063**

**Class and Section BSCS-7B**

**Department of Computer Science**

**BAHRIA UNIVERSITY, ISLAMABAD**

**Lab # 4: Uninformed Search in Python**

**Objectives:**

To implement the concepts of Uninformed search in python.

**Tools Used:**

Python IDLE 3.4

**Submission Date:**

**Evaluation: Signatures of Lab Engineer:**

**Task # 1:**

**Program:**

cost=0

class DWGraph(WGraph):

def add\_edge(self, u, v, w):

self.adj[u] = self.adj.get(u, []) + [v]

self.weight[(u,v)] = w

def dfs(self, start, path=[]):

path = path + [start]

global cost

for n in self.adj[start]:

if n not in path:

print(n)

if n=="A":

cost=cost +1

path = path + [n]

path = path + ["SR"]

self.dfs(n,path)

elif n=="B" or n=="C":

path = path + [n]

cost=cost+5

path = path + ["SR"]

self.dfs(n,path)

return path

def dls(self, start,goalnode,depth):

if depth==0:

if start==goalnode:

return goalnode

elif depth >0:

if start==goalnode:

return goalnode

elif start !=goalnode:

for n in self.adj[start]:

found=self.dls(n,goalnode,depth-1)

if found ==goalnode:

return found

def IDDFS(self,start, goalnode, maxDepth):

for i in range(maxDepth):

if (self.dls(start, goalnode, i)):

print(start)

return True

return False

a=DWGraph()

a.add\_node('SR')

a.add\_node('A')

a.add\_node('B')

a.add\_node('C')

a.add\_edge('SR','C',5)

a.add\_edge('SR','B',5)

a.add\_edge('C','B',5)

a.add\_edge('SR','A',5)

p=a.dfs('SR')

print(p)

print('cost is',cost)

pp=a.dls('SR','C',2)

print("goal node is",pp)

ppp=a.IDDFS('SR','A',4)

print("gpal node is",ppp)

**Result/Output:**

