**QUIZ 9**

Source Code:

class Graph():

def \_\_init\_\_(self, vertices):

self.graph = [[0 for column in range(vertices)]

for row in range(vertices)]

self.V = vertices

def isSafe(self, v, pos, path):

if self.graph[ path[pos-1] ][v] == 0:

return False

for vertex in path:

if vertex == v:

return False

return True

def hCycleUtil(self, path, pos):

if pos == self.V:

if self.graph[ path[pos-1] ][ path[0] ] == 1:

return True

else:

return False

for v in range(1,self.V):

if self.isSafe(v, pos, path) == True:

path[pos] = v

if self.hCycleUtil(path, pos+1) == True:

return True

path[pos] = -1

return False

def hCycle(self):

path = [-1] \* self.V

path[0] = 0

if self.hCycleUtil(path,1) == False:

print ("There is no Hamiltonian Cycle in the graph\n")

return False

self.printSolution(path)

return True

def printSolution(self, path):

print ("Solution Exists: Following",

"is one Hamiltonian Cycle")

for vertex in path:

print (vertex, end = " ")

print (path[0], "\n")

#Test case 1:

g1 = Graph(4)

g1.graph = [ [0,1,1,1],[1,0,1,1],[1,1,0,1],[1,1,1,0]]

g1.hCycle();

#Test case 2:

g2 = Graph(12)

g2.graph = [ [0,1,0,0,1,0,0,0,0,0,0,0],

[1,0,0,0,0,0,1,1,0,0,0,0],

[0,1,0,1,0,0,0,1,0,0,0,0],

[0,0,1,0,0,0,0,0,1,0,0,0],

[1,0,0,0,0,1,0,0,0,1,0,0],

[0,0,0,0,1,0,1,0,0,0,1,0],

[0,1,0,0,0,1,0,1,0,0,0,0],

[0,1,1,0,0,0,1,0,1,0,0,0],

[0,0,0,0,0,0,0,1,0,0,0,1],

[0,0,0,0,1,0,0,0,0,0,1,0],

[0,0,0,0,0,1,0,0,0,1,0,1],

[0,0,0,0,0,0,0,0,1,0,1,0] ]

g2.hCycle();

Output:

**Both test cases:**

harshavaidhyam@Harshas-MacBook-Pro quiz 9 % cd /Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design/quiz\ 9 ; /usr/bin/env /usr/local/bin/python3 /Users/harshavaidh

yam/.vscode/extensions/ms-python.python-2022.16.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 54743 -- /Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo

\ Design/quiz\ 9/quiz9.py

**One Hamiltonian Cycle**

**0 1 2 3 0**

**There is no Hamiltonian Cycle in the graph**