10

Department of Computer Science & Engineering
University of Asia Pacific (UAP)

Class Test #3 | Fall 2023 | 2nd Year 1st Semester

Course Code: CSE 203 | Course Title: Data Structures & Algorithms I | Credits: 3

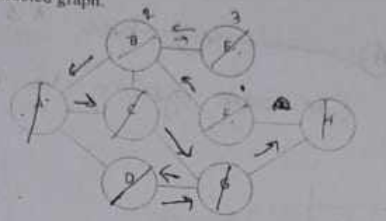
Full Marks: 10 | Duration: 30 minutes

Set 2

Name: Nusrat Nafiza

ID: ~~22201255~~ 22201253

1. Here an undirected graph. [2+2+6]



(a) Draw the adjacency matrix.
(b) Draw adjacency list.
(c) Apply an appropriate algorithm to transverse the graph **which uses recursion/stack. Your root node is ((last two digit of your registration number** % 8) + 1) *th letter in the alphabet.*

~~A→B,C,D~~
~~B→A,C,E,F~~
~~C→A,B,G~~
~~D→A,G~~
~~E→B~~
~~F→B,H~~
~~G→C,D,F~~
~~H→F,G~~

FBEACGD

8)53(6
48
5
+1
6

6

(c)

6th letter in the Alphabet is F.
so F is the root node. and we are going to use DFS · that use stack Data structure which follow LIFO (Last In first out)

Step: 1: Initially · stack and traverse list is empty. ~~Now~~ our root node is F. ~~F has two un visited~~ · · push F in stack and F has two unvisited

node B, H. ~~[illegible]~~ Now ~~push B, H~~. we going to push B.

traverse: F B

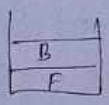
Step: 2 B has 3 adjacent node A, C, F, we are going to push E in

stack

traverse: F B E

step 3: Now move to E

E has no adjacent node so we are going to pop out.



pop: E

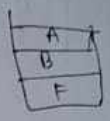
traverse: F B E

Step 4: Then we are going to back B Because our last element in stack is B. B has two unvisited node A, C, we are going to push

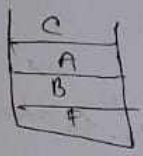
step 5: B A into stack mark it as visited

traverse: F B E A

pop: E



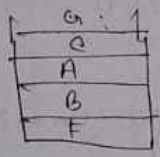
Step 6: Now A has two unvisited node C, D. we are going to push C in stack



traverse: F B E A C

pop: E

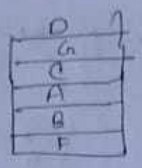
Step: 7: C node has one unvisited node G. we are going to push G in stack. and mark it as visited



traverse: F B E A C G

pop: E

step 8: a has two unvisited node H, D we are going to push D in stack.



traverse: FBEACGD

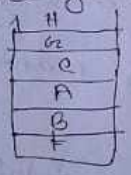
pop: E

Step 9: Now D has no unvisited node so we are going to pop out D.

traverse: FBEACGD

pop: E D

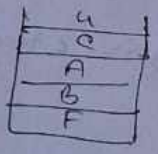
Step: 10: we are going to back G. and G has one unvisited node H. and we are going to push it in stack



traverse: FBEACGD H

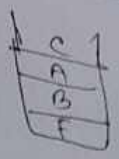
pop: E D

Step: 11 Now H has no 'Unvisited' node we are going to pop out it

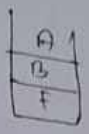


traverse: FBEACGDH

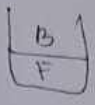
pop: E D H

Step 12: 

popout: EDHGC

Step 13: 

popout: EDHGCC

Step 14: 

popout: EDHGCCA

Step 15: 

popout: EDHGCCAB

Step 16: 
popout: EDHGCCABF

(a)

	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	0	0
B	1	0	1	0	1	1	0	0
C	1	1	0	0	0	0	1	0
D	1	0	0	0	0	0	1	0
E	0	1	0	0	0	0	0	0
F	0	1	0	0	0	0	0	1
G	0	0	1	1	0	0	0	1
H	0	0	0	0	0	0	1	1	0