UNIVERSITY PARTNER
UNIVERSITY OF WOLVERHAMPTON

iAcademy
Quality Education Improving Lives

HERALD COLLEGE KATHMANDU

Academic Year	Module	Assessment Number	Assessment Type
S20	Introductory Data Structures and Algorithms (DipIT02)	A1	Assignment Submission

[Assignment Submission]

Student Id : [NP03A190299]
Student Name : [Yogesh Shrestha]
Section : [DC8]
Module Leader : [Mr. Prakash Gautam]

Submitted on : 06-03-2020

Tutorial - 9.

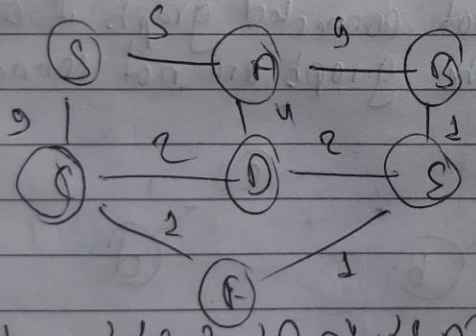
Page No.
Date : / /

1) Maximum number of possible non zero values in an adjacency matrix of a simple graph with n vertices are $n(n-1)$ or $n(n-1)/2$.

2) BFS graph traversal algorithm use a queue to keep track of vertices which need to be processed.

3) 400 boolean values will be needed to represent a using an adjacency matrix.

5)



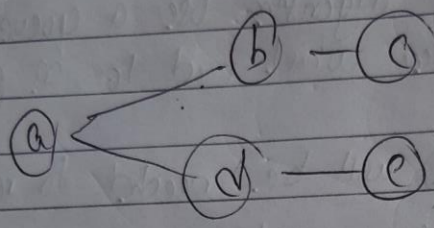
Edges = {(S,A), (A,B), (S,D), (A,D), (C,D), (D,E), (D,F), (B,E), (C,F), (F,E)}

$|E| = 10$

Page No.
Date : / /

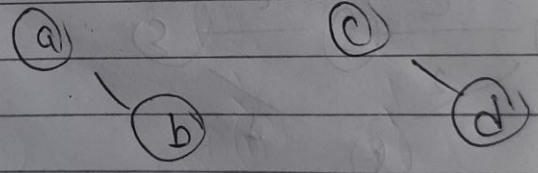
4 i) Connect Graph

A connected graph is a graph in which it's possible to get from every vertex in the graph to every other vertex through a ~~series~~ series of edges

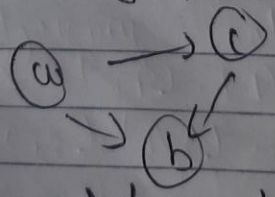


ii) Not connected graph.

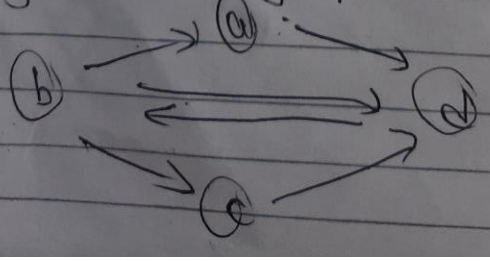
A graph is not connected graph if at least two vertices of the graph are not connected by a path.



iii) Weakly Connected graph

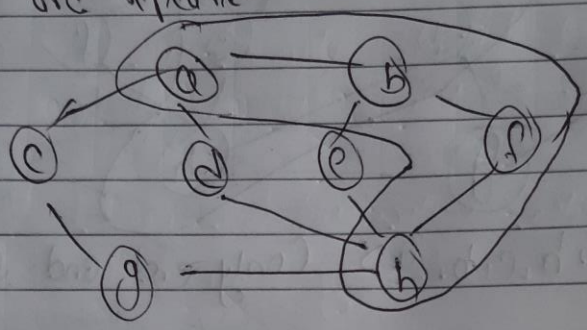


iv) Strongly connected graph



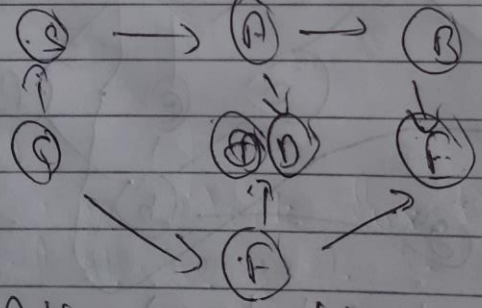
Page No.
Date : / /

(iii) Simple path
A walk in which no vertices and thus no edges are repeated.



< a, b, f, h >

6 (i) -> Solution.



For Adjacency matrix
Edges = { (A, B), (A, D), (B, E), (C, S), (C, F), (D, C), (D, E), (F, D), (F, E), (S, A) }.

Page No.
Date : / /

Now.

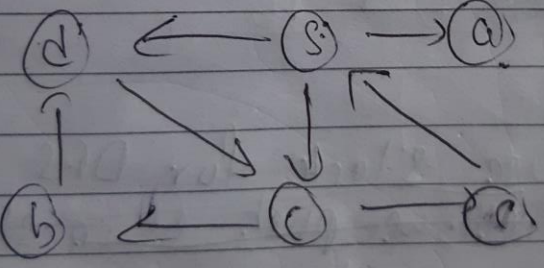
Vertex	List
0	A
1	B
2	C
3	D
4	E
5	F
6	S

Adjacency matrix

	A	B	C	D	E	F	S
A	0	1	0	1	0	0	0
B	0	0	0	0	0	0	0
C	0	0	0	0	1	1	1
D	0	0	1	0	0	0	0
E	0	0	0	0	0	0	0
F	0	0	1	1	1	0	0
S	1	0	0	0	0	0	0

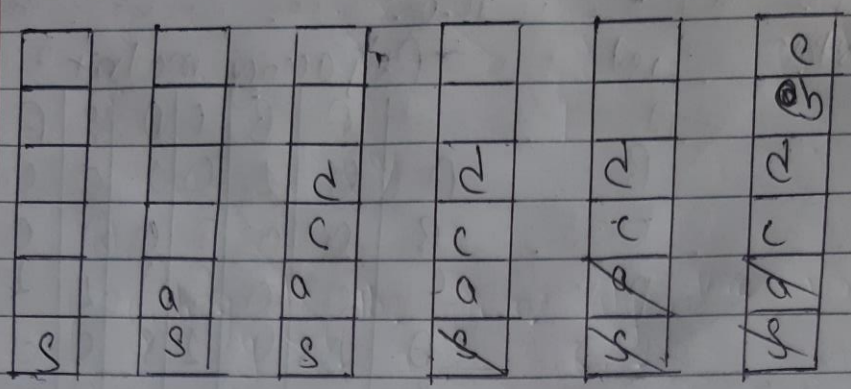
We use 1 to determine weight of edge or edge exist between them.

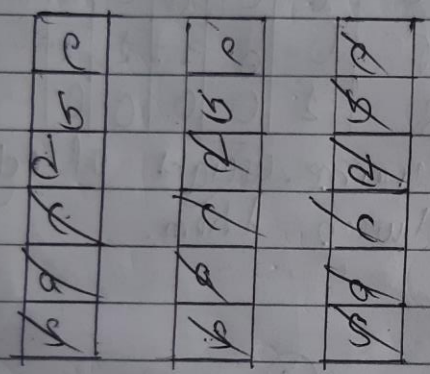
9. Sol.



(a) For BFS
We use array for BFS (Queue)

Page No.
Date : / /

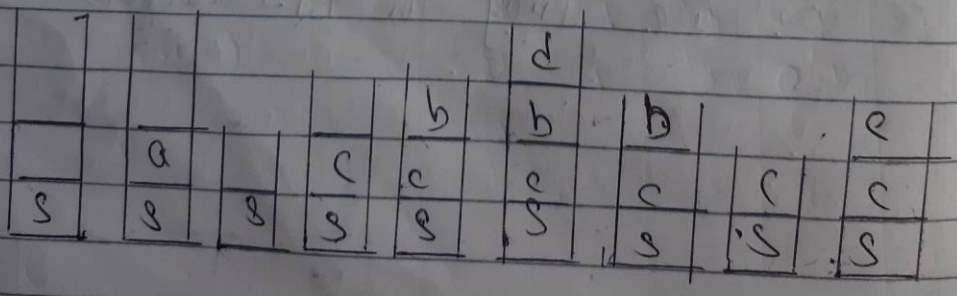




b) For DFS.

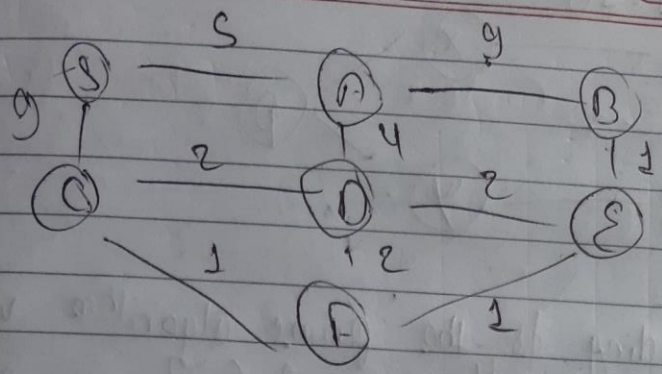
We use stack for DFS

~~first of all we push s' to stack~~



Page No.
Date : / /

5)

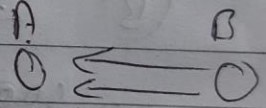


∴ The edges are : {{S,C}, {A,B}, {S,A}, {A,D}, {C,D}, {D,F}, {D,E}, {B,E}, {C,F}, {F,E}}.

∴ $|E| = 10$ #

8) Soln.

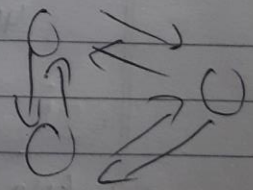
When $n = ?$



max possible $= ? \Rightarrow ? * 1$

When $n = 3$

max possible $= 6 = 3 * 2$

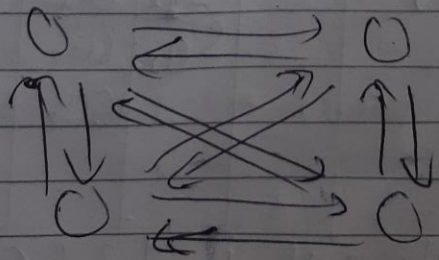


When $n = 4$

max. possible $= 12 = 4 * 3$

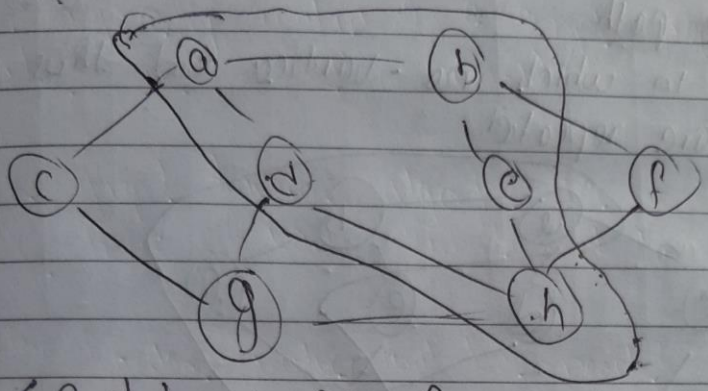
When $n = n$

max. possible $= \{n * (n-1)\}$.



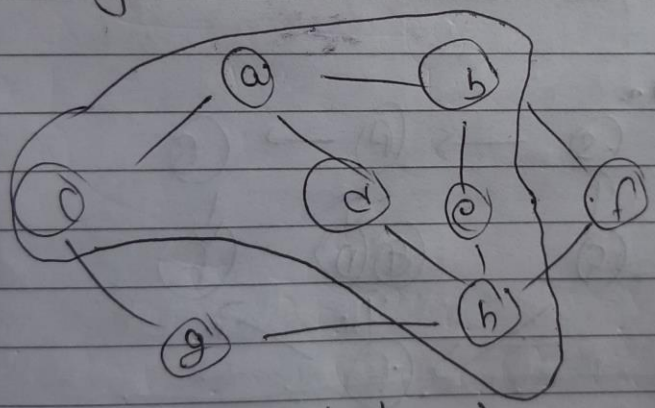
Page No.
Date : / /

v) Simple cycle



< a, d, h, e, b, a > (only end and start are same)

vi) closed cycle (w.r.t)



< c, a, b, e, h, d, a, c >

vii) ~~Dense~~ Dense graph
Graph is called dense if number of edges in the graph is ~~not~~ close to maximum possible number of edges