UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 212 Tutorial 7: Version A

- 13/04/2012
- 50minutes
- 2 questions for a total of 35 marks.

Name:____

Student/staff Nr:________________________________

Marker (office use):______________________________

Question 1 Graphs . . . . (23 marks)

1.1 For all of the following questions, if there is a choice between vertices to be processed next, process them alphabetically. Consider the graph in figure 1 and answer the following questions:

a) Give the adjacency matrix for this graph. List vertices alphabetically were applicable. (5)

Answer:

Solution: 1/2 mark for every correct row entry

	A	B	C	D	E	F	G	H	I	J
A	0	0	0	0	0	0	0	0	0	0
B	0	0	1	0	1	0	0	0	1	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	2	0	0
E	0	0	0	0	0	0	0	0	0	0
F	0	1	0	0	0	0	0	0	0	0
G	1	0	0	0	1	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	1	0	0	0	0
J	0	1	0	0	0	0	0	0	0	0

b) Assume the graph in figure 1 was undirected, give the incidence matrix for the graph in figure 1. Use the labels assigned to each edge, that is e_i for $i = 1..N$ (5)

Answer:

Solution: 1/2 mark for every CORRECT row. NOTE that the column indices could have been the vertices and the rows the edges. The solution here is the way the textbook does it.

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
A	1	0	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	0	1	0	0
C	0	0	0	0	1	0	0	0	0	0
D	0	0	0	0	0	0	0	0	1	1
E	0	1	0	0	0	0	0	1	0	0

F	0	0	1	0	0	0	1	0	0	0
G	1	1	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	1	1
I	0	0	0	0	0	1	1	0	0	0
J	0	0	0	1	0	0	0	0	0	0

c) The graph in figure 1 can be denoted as $G = (V, E)$. Draw the subgraph $G' = (V', E')$ of the graph G such that $E' = E$ and $V' \subset V$, $V' \neq V$ (2)

Answer:

Solution: Its the entire graph as given to them. Deduct 1 mark for every difference from the given graph (obviously only down to 0)

1.2 Why is it not sufficient when coding your own graph structure to have a graph class with a single reference to one of the vertices in the graph? (2)

Answer:

Solution: Need access to all vertices because graph may have disconnected components or if it is directed some vertices might not be reachable from every other vertex. There is also no particular "root" node in a graph.

1.3 Special care needs to be taken when using an incidence matrix for digraphs. Discuss this issue and the potential workarounds. (3)

Answer:

Solution: The issue is of where the edge originates (starting vertex as it were) and where it terminates (end vertex). This needs to be indicated in the matrix.

Could use integers as entries then 0 means no connection, 1 indicates starting vertex and 2 ending vertex (or some such answer)

OR

If edges are labelled as edge(v,w) then v indicates the starting vertex and w the ending one, in essence one could use the name of the edge to determine.

1.4 In terms of edges, what is the implication of removing a vertex from a graph? (2)

Answer:

Solution: Edges cannot be linked to only one vertex. All incident edges need to be removed as well.

1.5 Which graph representation would be most convenient to use if vertices and/or edges are removed on a regular basis? Motivate your answer. (2)

Answer:

Solution: Adjacency list (as a linked list). Removal and addition into these lists become linked list operation where lists grow and shrink conveniently.

NOTE: Other representations require matrices which will need to be recreated from every vertex removed/added.

1.6 Discuss the relationship between $|E|$ and the sum of the degrees of all vertices. (2)

Answer:

Solution: —E— is equal to half of the summed degrees of all vertices, i.e. $|E| = \frac{SummedDegrees}{2}$. Each egde contributes 1 to the degree of both the vertices it connects (i.e. adding 1 edge increases the summed degrees by 2)

Question 2 Graph Traversals . . . . (12 marks)

2.1 Graphs can be traversed with similar algorithms used for trees. What augmentations must be made to basic tree traversal algorithms such that they can be applied to graphs. Discuss the reasons for each augmentation. (4)

Answer:

Solution: 1. Graphs can be disconnected or not all vertices are always reachable from all other vertices, therefore we need an additional function/loop that will look for vertices that cannot be reached from the first vertex in the traversal.

2. Need some mechanism of indicating vertices as visited (flags or numbers or whatever) to prevent infinite traversals.

2.2 For breadth first search in graphs it is important to label vertices as visited (num(u) = i++) `before` they are inserted into the queue. Why is this? (2)

Answer:

Solution: If not then there is the danger of adding vertices which are already in the queue again into the queue. Nodes in the queue will be visited anyway and indicating them as visited will then mean they are already in the queue and should not be considdered for enqueue again.

2.3 Consider the graph in figure 1 and give the order in which vertices will be visited if depth first search were applied to this graph. If there is a choice between vertices to be processed next, process them alphabetically. (5)

Answer:

Solution: A,B,C,E,I,F,D,H,G,J

Break the sequence up as:

A,B 1 mark C,E 1 mark I,F 1 mark D,H 1 mark G,J 1 mark

2.4 What is the maximum number of forward edges that a graph may have? (1)

Solution: $|V| - 1$, One less than the number of vertices.

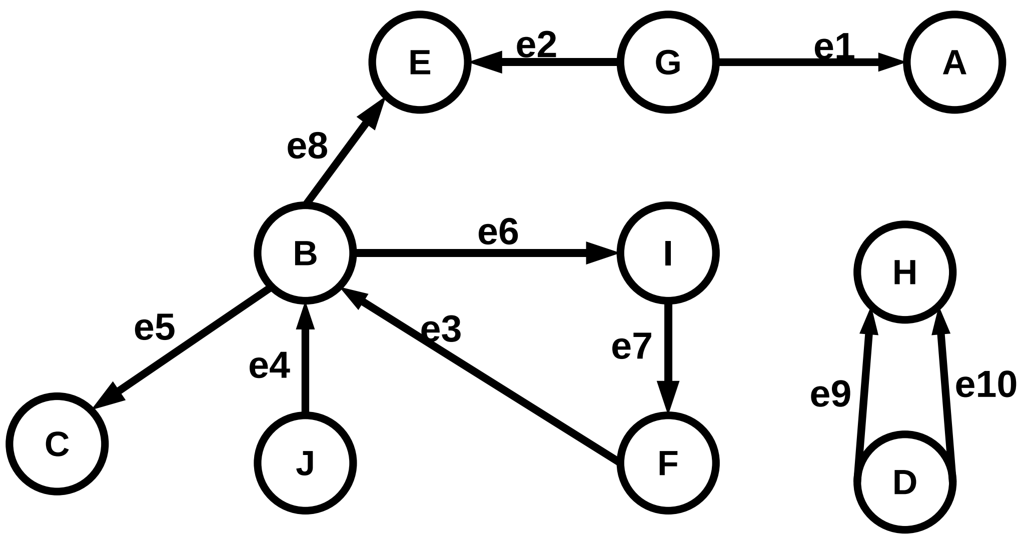

Figure 1: Graph