UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 212 Tutorial 7: Version B

- 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 Consider the graph in figure 1. Give the adjacency list representation of this graph. List vertices alphabetically were applicable. (5)

Answer:

Solution: 1/2 mark for every vertex having its correct list: NOTE: If they drew the adjacency list as a linked list thats also fine.

```
A -> G,H
|
B -> D,G,I
|
C -> B
|
D ->
|
E -> B
|
F ->
|
G -> E
|
H ->
|
I -> F
|
J -> B
```

1.2 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	1
B	0	0	1	1	1	1	0	1	1	0
C	0	0	0	0	1	0	0	0	0	0
D	0	0	0	0	0	0	0	0	1	0
E	0	1	0	0	0	0	0	1	0	0
F	0	0	0	0	0	0	1	0	0	0
G	0	1	1	0	0	0	0	0	0	1
H	1	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	1	1	0	0	0
J	0	0	0	1	0	0	0	0	0	0

1.3 If you are using an adjacency list implementation for a graph structure, how would you construct a weighted graph? (2)

Answer:

Solution: Create a node class with a member that represents the weight.

OR

Have a list/array representing the nodes and containing the weights

OR

Anything sensible that the student has written.

1.4 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.

1.5 Assume a graph G with $|E| = 41$ and a vertex B where $deg(B) = 7$, what will $|E|$ be after the removal of B? (1)

Answer:

Solution: 34

1.6 Which graph representation would be most convenient or efficient to use if vertices are never added or removed once the graph has been created but edges are added and removed on a regular basis? Motivate your answer. (2)

Answer:

Solution: Adjacency matrix. You simply update the values in the matrix using vertices as indices. Array accesses are cheap compared to having to maintain linked lists.

NOTE: Incedency matrices will have to be resized dynamically if edges are added (or removed). Lists are cumbersome to delete and insert out of and whole nodes as opposed to array element updates will have to be deleted/inserted.

1.7 Consider a simple graph $G = (V, E)$ with $|E| = 7$. What is the smallest possible value for $|V|$? (2)

Answer:

Solution: 8, each egde connects 2 vertices. Isolated vertices will not increase the edge count.

1.8 Draw the graph K_5 (2)

Answer:

Solution: There needs to be 5 vertices. Each vertex is connected with EXACTLY one edge to every other vertex, i.e. there cannot be more than one edge between two vertices.

1 mark for the number of vertices correct. 1 mark for sticking to the number of edges between vertices.

1.9 What is the number of edges $|E|$ in the graph K_{101} (1)

Answer:

Solution: $|V|^2$ so that is $101^2 = 10201$

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

2.1 Graphs can be traversed with similar algorithms as 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 breadth 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,G,H,E,B,D,I,F,C,J

Break sequence up as: A,G 1 mark H,E 1 mark B,D 1 mark I,F 1 mark C,J 1 mark

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

Solution: 0 (It is possible that the graph consists of only isolated vertices and no edges)

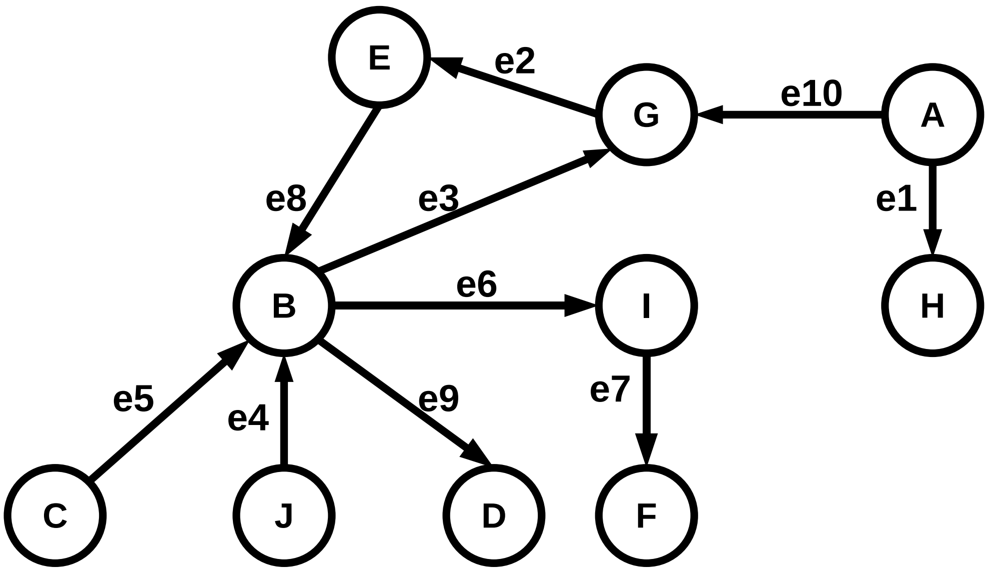

Figure 1: Graph