27. Given a drawing of a directed graph along with the "discovery" and "finish" times of its vertices after a depth-first search, write the labels of the graph in topological order. Explain how you obtained the ordering.

28. Given a drawing of a directed graph along with the "discovery" and "finish" times of its vertices after a depth-first search, identify the type of each edge (tree, back, forward, or cross). The following test is relevant, with `d[v]` the discovery time of vertex `v` and `f[v]` its finish time:

```
if (d[v1] = d[v2] - 1)
   (v1,v2) is a tree edge
else if (d[v1] > d[v2]  && f[v1] < f[v2])
   (v1,v2) is a back edge
else if (d[v1] > d[v2] && f[v1] > f[v2])
   (v1,v2) is a cross edge
else //  d[v1] < d[v2] - 1  && f[v1] > f[v2]
   (v1,v2) is a forward edge
```

Disjoint Set

29. Give the value of the expression $lg^*(1024)$, where $lg^*(N)$ is the iterated base-2 logarithm of N.

30. Write pseudo-code to find the connected components of an undirected graph $G = (V, E)$ using Union-Find operations on the vertices.

31. Define *Union-by-Weight heuristic*.

32. Define *Path-compression heuristic*.

33. When both the union-by-weight and the path-compression heuristics are applied on disjoint sets totaling N items, a sequence of M union-find operations can be done in $O(M \lg^*(N))$ time. It is sometimes said that under these conditions, union-find is done in constant time per operation. What does this mean? Why is it true?

34. In an uptree with root x, let $R(x)$ be the length of the longest path and let $N = W(x)$ be the number of vertices (including x). Assume the uptree was created by means of multiple applications of the "union-by-size" algorithm.
Prove: $R(x) \leq \lg(N)$

20. Draw the directed graph represented by the adjacency list given below.

```
v[1] (Label = A) --> 2 --> 5

v[2] (Label = B) --> 3 --> 5

v[3] (Label = C) --> 2 --> 4 --> 5

v[4] (Label = D) --> 4 --> 5

v[5] (Label = E) --> empty
```

21. Discuss the characteristics of the *adjacency table* and *adjacency list* implementations of graphs. Include storage requirements and asymptotic worst-case performance of the operations:

```
Note:  u and v are vertices in the graph

  Degree(u)       returns the degree of vertex u  (undirected graphs)
  InDegree(u)     returns the indegree of vertex u (directed graphs)
  OutDegree(u)    returns the outdegree of vertex u (directed graphs)
  AdjacentTo(u)   returns a list of the vertices adjacent to u
  AdjacentFrom(u) returns a list of the vertices adjacent from u
  Connected(u,v)  returns true if there is an edge between
                      vertices u and v, returns false otherwise
```

22. Consider the directed graph represented by the following adjacency matrix:

```
  |  A B C D E
--+------------
A |  0 1 1 1 1
B |  0 1 1 0 0
C |  0 0 0 1 0
D |  0 0 0 1 1
E |  0 1 0 0 0
```

List the depth-first and breadth-first traversals of the graph beginning at vertex A. Repeat for vertex B. (Whenever a new vertex is to be visited and there is more than one possibility, use the vertex labelled with the letter than comes first in the alphabet.)

23. Define *directed acyclic graph*.

24. Define *topological ordering* of a directed acyclic graph.

25. Given a drawing of a graph, find all cycles.

26. Given a drawing of a directed acyclic graph, write the labels of its vertices in topological order. Explain how you obtained the ordering.

Graphs

9. Define *graph*, *undirected graph*, *directed graph*, and *weighted graph*.

10. Define *path* in a graph. Define *length of a path* in a graph.

11. Define the following:

    (a) Connected, undirected graph.

    (b) Strongly connected directed graph.

    (c) Weakly connected directed graph.

12. Let $G = (V, E)$ be an undirected graph with V the set of vertices and E the set of edges. Let $v_1, v_2, \ldots, v_p \in V$ be the members of V and let $q = |E|$ be the cardinality of E. Prove:

$$\sum_{i=1}^{p} \text{degree}(v_i) = 2q$$

13. Prove that in any undirected graph, the number of vertices of odd degree is even.

14. Write pseudo-code for breadth-first and depth-first traversals of undirected graphs. The code must be complete and must fully describe the operations.

15. Describe, in English, any *adjacency table* graph implementation. How does the implementation differ for directed and undirected graphs?

16. Describe, in English, any *adjacency list* graph implementation. How does the implementation differ for directed and undirected graphs?

17. Given a drawing of a directed or of an undirected graph, show its representation as an adjacency matrix.

18. Draw the directed graph represented by the adjacency matrix given below. The rows/columns in the matrix correspond to the vertices with labels A,B,C,D,E.

```
0 1 0 1 0
1 1 1 0 0
0 0 0 0 1
0 1 0 0 1
0 0 0 0 0
```

19. Given a drawing of a directed or of an undirected graph, show its representation as an adjacency list.

DRAFT - WILL CHANGE

CMSC–341 Data Structures
Spring 1999 2 May 1999 Final Review

The exam will be drawn entirely from the following questions. There will be no "surprise" or "trick" questions on the exam. Every question on the exam is in this handout except that some questions may be phrased differently or involve different data.

Exam 1 Questions

1. Every question from Exam 1. From the exam itself, **not** from the exam review questions.

Exam 2 Questions

2. Every question from Exam 2. From the exam itself, **not** from the exam review questions.

B Trees

3. Define B-Tree.
4. Give pseudo-code for search in a B-Tree of order M.
5. Given a drawing of a B-Tree, show the tree after insertion of a given element.
6. Given a drawing of a B-Tree, show the tree after deletion of a given element.
7. Draw a B-tree with M = 4 and L = 3 containing the integer values 1 to 25.
8. Given some characteristics of an external storage problem:
   (a) The number of items to be stored.
   (b) The size (in bytes) of the key for each item.
   (c) The size (in bytes) of each item.
   (d) The block size (in bytes) of a disk block.

   design a suitable B-tree (give its order, M and leaf size, L).