CMSC 341 Data Structures Final Exam Review

These questions will help test your understanding of the material discussed in class, textbook, and homework and project assignments. These questions are only a study guide. Questions found here may be on your exam, although perhaps in a different format. Questions NOT found here may also be on your exam.

Final exam is not accumulative and will cover the data structures of Thread, Disjoint Set, and Graphs. There will be a question that asks you about all data structures you have learned this semester. Be prepared.

Thread

1. What is a thread and what are the differences between threads and processes?
2. What are the two ways to define and start a thread?
3. Why do we need interruption in thread programming and how to implement it?
4. What are intrinsic locks and synchronization?
5. What does it mean for a program to be thread safe? We talked about one way to do it in class – what is it and how to implement it?
6. Describe an example of code that is not thread safe and indicate how to make it thread safe.

Disjoint Sets Review

1. Define $lg^*(N)$. What is the value of $lg^*(1024)$?

2. Define the Union-by-Weight heuristic.

3. Define the Path Compression heuristic.

4. When both Union-by-Weight and Path Compression are used on disjoint sets with a universe of N elements, 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?

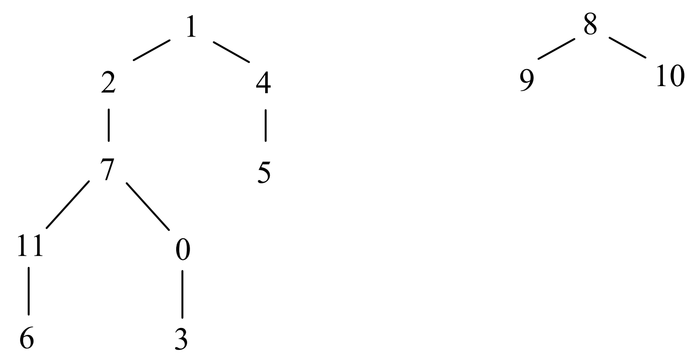
5. In an uptree with root x, let $R(x)$ be the length of the longest path and let N be the number of nodes (including x). Assuming the uptree was created by means of multiple union operations using the Union-by-Weight heuristic. Prove $R(x) \leq lgN$.

6. Perform the following Union-by-Weight operations on a universe of 10 elements (0-9, each initially in their own set). Draw the forest of trees that result. U(1,5); U(3,7); U(1,4); U(5,7); U(0,8); U(6,9); U(3,9). If the sets have equal weight, use the root with the smaller value as the root of the new set.

7. Although uptrees are used to conceptualize disjoint sets, disjoint sets are generally implemented in an array. Explain how this is possible.

8. Prove that if Union-by-Weight is used for all unions, the length of the deepest node is no more than *lg*(*N*).

9. Given the following forest of uptrees,
a. show the array which represents them
b. show the result of find(6), using Path Compression



Graphs Review

1. Define the following terms
    a. Graph
    b. Weighted Graph
    c. Directed Graph
    d. Undirected Graph
    e. Path
    f. Length of a Path
    g. Sparse Graph
    h. Dense Graph
    i. Connected Undirected Graph
    j. Weakly Connected Directed Graph
    k. Strongly Connected Directed Graph
    l. Adjacency Matrix
    m. Adjacency List
    n. Directed Acyclic Graph
    o. Topological Ordering
    p. Cycle

2. Let G = (E, V) be an undirected graph. Let v1, v2, v3… vp be the members of V, and let q = |E| (the cardinality of E). Prove that the sum of the degrees of all the vertices is equal to 2q.

3. Write pseudo-code for the breadth-first and depth-first traversals of an undirected graph.

4. Given the drawing of a graph, list the breadth-first and depth-first traversals of the graph.

5. Describe, in English, an adjacency matrix graph implementation. How does an adjacency matrix differ for directed and undirected graphs?

6. Describe, in English, an adjacency list graph implementation. How does an adjacency matrix differ for directed and undirected graphs?

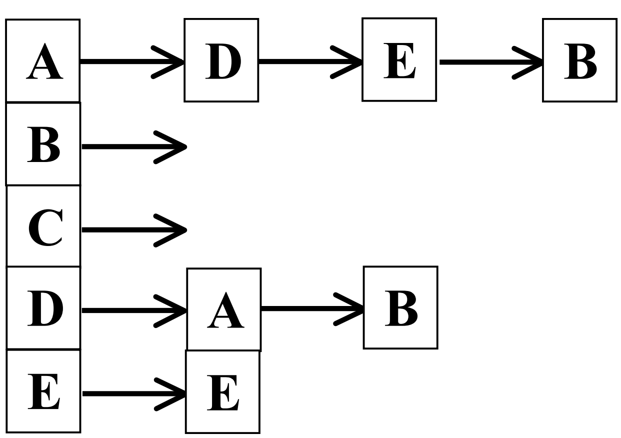
7. Given the drawing of a directed or undirected graph, show its representation in an adjacency matrix or adjacency list.

8. Draw the weighted directed graph represented by the adjacency matrix below. A non-zero value at [row, column] indicates that the vertex in the row is adjacent to the vertex in the column:

	A	**B**	**C**	**D**	**E**
A	0	5	8	0	0
B	3	0	6	0	0
C	0	3	4	1	0
D	0	6	7	0	0
E	0	0	0	0	0

9. Given the drawing of a(n) (un)directed graph, show its representation in an adjacency list.

10. Draw the directed graph represented by the adjacency list below. Each element in a vertices list is adjacent to the vertex.



11. Given the drawing of a graph, find all cycles (see examples in class).
12. Discuss the characteristics of the adjacency matrix and adjacency list implementations for a graph. Include storage requirements and worst-case performance for all graph operations.
13. Given a directed graph whose edges have positive weights, use Dijstrka's algorithm to find the shortest path between a given source and destination.
14. Explain why Dijstrka's algorithm only works for graphs whose edges have positive weights.