All topics from Module 1 through and including Module 4

Review assignments 1, 2 and 3.

In general, be able to write algorithms discussed, determine/explain runtime, prove or justify correctness, explain how algorithms work, give an example of the algorithm.

Write algorithms that may be a variation of those reviewed in class and

- Explain the main concept of your algorithm
- Give pseudo-code
- Present an example of running your algorithm
- Prove/justify its correctness and its running time

Given input, provide the output and/or other information relating to the algorithm.

- e.g. for graph algorithms, given a representation of a graph, provide info on the DFS tree, BFS tree, Directed DFS or BFS, Top Sort order and other information that may be derived from the algorithm

Proof techniques

- Proof by Induction
- Proof by contradition
- Proof by counter example
- Proof by explaning a loop invariant

show a recursion tree for runtime analysis

What does Big-Oh, Big-Theta, Big-Omega, little-oh, little-theta and little-omega mean?

- Given function function f(n) use limit rule or defintion of Big-Oh, Big-Theta, … to show that it is in g(n)
- Given some code, determine the runtime

Algorithms Discussed

- arrayMax
- recursiveArrayMax
- iterativeFactorial
- recursiveFactorial

Sorting Algorithms

- Merge Sort
- Quick Sort
- Heap Sort
- Bucket Sort
- Radix Sort

What is the selection problem and how to solve, runtime analysis.

What is Median-of-Medians, when used, why important, runtime analysis.

Union Find Connected Components Algorithm and the Union Find Data Structure made with linked-lists

- operations (a.k.a. functions)
- runtime analysis for different operations
- runtime for creating the data structure

Graphs

- terminology for directed, undirected, acyclic graphs
- how to represent internally (adjacency lists via vertex or edges, adjacency matrix)
    - -- explain/give runtime of operations on graphs that are represented by an adjacency list or adjacency matrix
- Graph Algorithms
    - -- Depth First Search (DFS)

-- Breadth First Search (BFS)

-- Directed DFS

-- Directed BFS

-- Topological Ordering