1. Concepts behind $O$, $\Omega$, and $\Theta$. Know what they are good, their definitions, and how to show that a certain expression involving the size of a problem, usually specified as a function of $n$ is bounded above, below, or both by another expression of $n$.

2. Heaps, both max and min heaps, and how they are implemented using an array. Remember the max and min heap properties, and how to add and extract values from a heap.

3. Definitions of a graph, an acyclic graph, a directed graph and a directed acyclic graph.

4. Definition and the properties of a bipartite graph.

5. Topological orderings, and how they relate to directed acyclic graphs.

6. Minimum spanning trees.

7. Dijkstra’s shortest path algorithm.

8. Proofs by induction. Must include the base case, the inductive hypothesis, and the inductive step.

9. Decomposing a problem into multiple subproblems

10. Recurrence relations for specifying value to be minimized or maximized in terms of these subproblems.

11. Optimal substructure

12. Greedy versus dynamic programming approach to designing algorithms.


15. Making change with coins


18. Fractional and 0/1 Knapsack problems.

19. Huffman codes.

20. Proofs by induction. Must include the base case, the inductive hypothesis, and the inductive step.

21. Classes of problems, P, NP, NP-Complete, EXP.

22. Polynomial-time Reduction

Consult our textbook, lecture notes and homework problems to find example problems for each topic. Study with a friend, and construct example questions for each other and discuss the answers. Test questions will be short answer, multiple choice, some proofs, and some will require drawing diagrams.