Note:

- The final will be a cumulative exam. But, for the chapters covered by the previous exams, most problems will be easy to answer.
- Final exam has 27% weight. For comparison, midterms 1 and 3 have weight 18%, and midterm 2 weight 17%.

Topics that have not been covered by the previous exams (about 70%)

- Ch 22. Running time of BFS and DFS. No need to study BFS implementation. Representing a graph by matrix, or adjacency lists; their pros and cons. After running DFS, can you label each edge as forward, back, tree, or cross edges? Running topological sort. Running the algorithm of finding strongly connected components. Component graphs. Understanding strongly connected components and component graphs; e.g. can a component graph have a cycle? Two key theorems: white path theorem and parenthesis theorem. Showing the correctness of topological sort. How to see if a directed graph has a cycle or not; prove the correctness.
- Ch 23. Minimum spanning trees. Basic properties of trees. Kruska's and Prim's. Illustrating the algorithms on a given instance. E.g. output the edges in which they are chosen. Notion of safe edges. Proving that a safe edge must appear in all MSTs. Proving the correctness of algorithms using the notion of safe edges. Why the heaviest edge on a cycle is not safe. Show that a given edge is safe: you should find a cut such that the edge is the lightest edge crossing the cut.
- Ch 24. Shortest paths from a single source. Bellman-Ford, for DAG, and Dijkstra. Illustrating the algorithms on a given instance. Their running time. Understanding each algorithm's assumption on the input. Why do they work? Intuitive understanding of the key lemmas/observation. Number of rounds in BF.
- Ch 26. Augment a flow along a path and show its resulting residual graph. RT of Ford-Fulkerson. How do you compute an s-t min-cut. Max flow = Min Cut. Application: Bipartite matching.

Topics that have been covered by the previous exams (about 30%)

- Ch 2. Not much to ask. Basic understanding of Insertion sort and Merge sort. RT. RAM. No question on Loop invariant or recurrence based proof.
- Ch 3. Asymptotic notations, comparing asymptotic quantities/functions. Properties of asymptotic notations.
- Ch 4. Basic understanding of Max sub array problem and Strassen's algorithm. Their RT.

- Ch 6. Heap. Basic operations of max heap. Their running time. Max priority queue and its operations. Their running time.
- Ch 7. Quicksort. A high-level understanding of the analysis of randomized quicksort.
- Ch 8. Decision tree for proving lower bound for all comparison based algorithms. Understanding what the decision tree means. Counting sort, Radix sort, and bucket sort. When do you use them? What are the assumptions the algorithms leverage to break the lower bound? Stable sort.
- Ch 9. Randomized Selection. No need to study deterministic algorithm for the selection problem. RT.
- Ch 11. Hashing. RT (under the assumption of simple uniform hashing)
- Ch 15. LCS.
- Ch 16. Key lemma for Interval Selection. Huffman algorithm.