The final exam will cover all material that we covered in class starting with the class immediately preceding the second exam. The major topics are listed below. You are expected to know the algorithms and their runtime complexities.

- Sorting
  - Quicksort
  - Heap sort
  - Lower bound for sorting by binary comparisons
  - Definition and use of decision trees
- Disjoint set ADT
  - Equivalences and the dynamic equivalence problem
  - Basic data structure
  - Smart union and path compression algorithms
- Graph Algorithms
  - Graph representations
  - Topological sort
  - Kruskal's minimum spanning tree
  - Dijkstra's single-source shortest paths
  - Floyd-Warshall all-pairs shortest paths
  - Depth-first search
- Complexity Classes
  - Definitions of P, NP
  - Examples of problems
  - NP-completeness
  - NP-hardness
  - Polynomial time reducibility

For each topic, performance analysis is part of the topic. The format of the exam includes true/false questions, short answer questions, and questions that ask you to analyse algorithm performance or carry out algorithms on examples.

Some sample questions of various types are below.

1. Draw a decision tree for the problem of sorting three numbers.
2. What is the least number of comparisons needed to sort an array of 6 numbers, in the worst case, using any sorting algorithm that sorts with binary comparisons?
3. Given a collection of disjoint sets and a given sequence of union and find operations, show the final state of the collection assuming union-by-size and/or path compression is used, and assuming that sets are represented by parent trees.

4. Find a minimum spanning tree for a given graph.
5. Write a pseudocode description of a topological sorting algorithm.
6. Write a pseudocode description of Kruskal's algorithm.
7. What is the asymptotic worst case running time of Kruskal's algorithm? Of Dijkstra's single source shortest path algorithm?
8. Find the shortest paths from the given vertex to all other vertices in the given graph.
9. Write out a topological sort of the given graph.
10. Give an example of an NP-complete problem.
11. What problem can be proved to be NP complete using polynomial reduction from Hamiltonian Circuit.
12. If someone discovers a polynomial time deterministic algorithm for the Hamiltonian Circuit problem, draw what that would imply as a Euler diagram showing the relationships between the sets P, NP, NP-Hard, and NP-Complete.