The exam will cover all material that we covered in class ***after*** March 15. The major topic categories after those are:

- Heaps and Priority Queues
- Sorting
  - Shell sort
  - Quicksort
  - Heap sort
  - Lower bound for sorting by binary comparisons
  - Running times of insertion and selection sort in terms of data moves and key comparisons
  - Definition and use of decision trees
- Disjoint set ADT
  - Equivalences and the dynamic equivalence problem
  - Basic data structure
  - Union-by-size and find using path compression algorithms
- Graph Algorithms
  - Graph representations
  - Topological sort
  - Kruskal's minimum spanning tree
  - Dijkstra's single-source shortest paths
  - Depth-first search
- Complexity Classes
  - Definitions of P, NP
  - Examples of problems
  - NP-completeness
  - NP-hardness

For the listed algorithms, you are expected to be able to do the following:

- Apply the algorithm to a particular input. The input might be represented in the exam with a visual depiction, such as a tree or graph, or by a data structure representing it.
- State the asymptotic worst case running time of the algorithm, and if we have covered it, the average and/or best cases for that algorithm. (For example, insertion, selection, and quicksort average and best cases were covered.) Running times should be known in terms of the size of the input.
- Give an example of an input for which it will lead to a worst case running time for all sorting algorithms.
- Write a pseudo-code description of the algorithm.

For data structures and representations, you should be able to do the following:

- Given one form of representation, convert it to another (adjacency matrix to adjacency list and vice versa), or picture as lines and nodes to either of these, and similarly for sets and parent trees and arrays, and heaps and their arrays representation.

The format of the exam includes true/false questions, short answer questions, and questions that ask you to analyse algorithm performance, carry out algorithms on examples, or write small chunks of code.

Some sample questions of various types are below. Answers are not provided.

1. Build a heap from an array containing the keys $30, 40, 25, 60, 75, 23, 86, 12, 72$ in that order. Show it as an array and as a binary tree.
2. True or False? The average number of comparisons to insert n elements into a binary heap is $O(n \cdot \log\, n)$.
3. True or False? The average running time of quick sort is $O(n \cdot \log\, n)$.
4. Draw a decision tree for the problem of sorting three numbers.
5. Given an array, show its state after shell-sorting with a gap sequence of gap $= 5$, then gap $= 3$ (or any other gap sequence of course).
6. 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?
7. 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.
8. Draw a minimum spanning tree for a given graph.
9. Write a pseudo-code description of a topological sorting algorithm.
10. Write a pseudo-code description of Kruskal's algorithm.
11. What is the asymptotic worst case running time of Kruskal's algorithm? Of Dijkstra's single source shortest path algorithm?
12. Find the weights of the shortest paths from a given vertex to all other vertices in the given graph.
13. Write out a topological sort of the given graph, either by numbering the nodes or listing them in the correct order.
14. Give an example of an NP-complete problem.
15. What problem can be proved to be NP complete using polynomial reduction from Hamiltonian Circuit.