# CSCI 335 Software Design and Analysis III
# Final Exam Topics and Sample Questions

The exam will cover all material that we covered in post-midterm: Lecture 14 and all following Lectures, the multiple choice in class tests and assignments 3. 4 and 5.

The major topic categories  are the following. For each topic, study what it is, why approach is useful, what applications it is useful for and its complexity where appropriate, the weaknesses of the approach :

- Heaps and Priority Queues
  - Binary Heap, Heap order property, operations
  - Selection Problem
  - Merging, d-heaps, leftist heaps, skew heaps, binomial queues
- Sorting
  - Insertion sort
  - Shell sort
  - Quicksort
  - Heap sort
  - Merge 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
  - Basic data structure
  - Find and union details
  - Role in Kruskal
  - Smart union, Path compressions
- Graph Algorithms
  - Graph representations
  - Topological sort
  - Single-source shortest paths
  - Network Flow
  - Minimum spanning tree
  - Depth-first search
- Algorithm Design
  - Greedy Algorithms
  - Divide and Conquer
  - Dynamic Programming

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.
- Lower bound proofs, telescoping proofs, worse case analysis, average-case analysis: Prove as shown in class or there could be what or why questions asked about some part of the proof shown in class.
- Apply different bin packing strategies for a given input.

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

Some sample questions of various types are below. The exam may or may not cover these exact questions and may have similar questions on other topics from the relevant material. 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. For each algorithm below, provide the time complexity. In graph algorithms provide the complexity as a function of # of vertices and #of edges: Merge Sort worst case, MST, Dijkstra using priority queue, DeleteMin in leftist heap with N items etc.
4. Draw a decision tree for the problem of sorting three numbers
5. Given an array, show its state after shell-sorting with increment h= 5, then h = 3 (or any other increment 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. Draw a minimum spanning tree for a given graph.
8. Write a pseudo-code description of a topological sorting algorithm.
9. Write a pseudo-code description of Kruskal's algorithm.
10. What is the asymptotic worst case running time of Kruskal's algorithm? Of Dijkstra's single source shortest path algorithm?
11. Find the weights of the shortest paths from a given vertex to all other vertices in the given graph.
12. Write out a topological sort of the given graph, either by numbering the nodes or listing them in the correct order.
13. Suppose the topological sort algorithm used a stack where vertices are alphabetical in the adjacency list using the  instead of a queue for graph in q13 , does a different ordering result? If yes, provide the new ordering. If no, “write no change”.

14. Provide TWO advantages of Shellsort over Mergesort and ONE disadvantage of shellsort over mergesort.
15. Running time for heap operations
16. Divide and Conquer vs Dynamic Programming
17. Where is the assumption of acyclic  graphs critical? Why?