1. Binary Search Trees
   (a) Binary Search Property
   (b) Operations on binary search trees
      i. Traversals: inorder, preorder and postorder
      ii. Breadth-First traversal
      iii. Search
   (c) Best case v.s. worst case performance

2. AVL Trees
   (a) HB[1] property
   (b) Re-balancing: single and double rotations
   (c) Operations on AVL trees: search, insert, delete
   (d) Best case v.s. worst case performance; comparison to binary search trees.

3. “Big O” notation.
   (a) Time complexity for operations on data structures
   (b) Analyzing iterative code / nested loops.

4. Heaps
   (a) Heap property
   (b) Implementation in dense arrays
   (c) heapify() operation
   (d) building a heap from an unordered array
   (e) heapsort

5. Tries
   (a) Use as a dictionary
   (b) Operations: insert, search, delete

6. Graphs
   (a) Adjacency matrices, adjacency lists.
   (b) Graph traversal with pre- and post- numbers.
   (c) Topological sort of a directed acyclic graph

7. Miscellaneous
   (a) Hash Tables: Load factor
   (b) Hash Tables: Average length of a chain, standard deviation of a chain.
   (c) Searching a collection of records by multiple keys
   (d) Abstract data type – what is it ?
   (e) Recursion
   (f) Huffman codes

8. One big-picture question. Short essay on how to use data structures effectively in a “real-world” problem/application.