Data structures, algorithms and code constructs you'll need for your interview

Throughout this course we've talked about several data structures and algorithms you'll need to have in your toolbox as you approach an interview. Here we're providing a list of the most important ones. You should probably know how to (1) use, (2) implement, and (3) analyze the running time of all of the data structures and algorithms listed below. This is not an exhaustive list by any means, and if you know of more, please post them in the forum. If the post gains enough momentum, we'll pin it to the top.

Most of these concepts are covered to some degree in the previous courses in our specialization, but it never hurts to find other sources and study them as much as possible. And don't try to "just memorize" anything. If your understanding of these concepts is solid, you'll be much better off in a high-stress situation!

Data Structures and Abstract Data Types

- Arrays
- Linked Lists: single and double, with and without sentinel nodes
- The List ADT
- Trees: Binary and otherwise, BSTs
- The Set ADT
- Tries
- Heaps
- Stack, Queue and Priority Queue ADTs
- Graphs: directed and undirected, different representations and their trade-offs
- Hashtables: collision resolution strategies, hash functions and their trade-offs
- The Map ADT

Algorithms

- Binary search

- Sorting algorithms: Selection Sort, Insertion Sort, Bubble Sort, Quicksort, Mergesort, Heapsort
- Search algorithms: Depth-first search, Breadth-first search, Dijkstra's algorithm
- Tree traversal: pre-order, post-order, in-order, level-order (which is BFS)
- Algorithm techniques: Brute force, Divide and conquer (e.g. Mergesort), Greedy, Dynamic Programming (more advanced, OK not to know if you are at an intermediate level)
- NP-hard problems

Basic Java Programming Constructs

- All the very basics of course (loops, conditionals, functions, etc)
- Input and output: from files and command line
- The String and Character class
- Converting between data types
- Object oriented design and concepts: Inheritance and Polymorphism, Classes and Objects, static and non static, access level modifiers (public, private, protected)
- The java.util library (including all of the container classes)
- The Iterator pattern (more advanced, probably OK not to know if you are at an intermediate level)
- Exception generation and handling
- Event-driven programming
- Inner classes and anonymous classes (slightly more advanced, but good to know about)
- Lambda expressions (more advanced)

Classes of problems to study

Several kinds of questions are standard in an interview setting. Make sure you seek practice with each of these different kinds of problems.

- Object oriented design, using given data structures.
- Bitwise manipulation of integer representations.
- Game theory puzzles: "muddy children", volcano eruptions.

- Estimating bounds: "how many chickens are there in Chicago?"