DSA Midterm Study Guide

Contents

The midterm will cover chapter 18-23

18. Recursion
19. Generics
20. Lists, Stacks, Queues, and Priority Queues
21. Sets and Maps
22. Developing Efficient Algorithms
23. Sorting

Suggested Study Methods

1. Review Quiz and Assignment Topics
2. Review "Key Point" notes from textbook
3. Review lecture slides and lecture notes
4. Review the suggested topic outline below
5. Study textbook examples

Suggested Topic Outline

1. Recursion
    1. Definition
    2. Fibonacci and Factorial Algorithms
    3. Know what tail recursion is
    4. Know what types of problems are served well by recursion
    5. Be able to compare and contrast recursion and iteration
    6. Know the benefits of recursion
2. Generics
    1. Understand the motivation and benefits of generics
    2. Be able to identify a generic method, generic class, interface, etc.
    3. Understand type erasure and its implications
3. Lists, Stacks, Queues, and Priority Queues
    1. Understand the purpose, benefits, and properties of each type of data structure
        1. Understand of any sorting properties of each data structure
        2. Understand how elements are added and removed from each data structure
    2. Be familiar with the methods
    3. Be familiar with the concrete types within each interface, e.g. ArrayList and LinkedList, and their properties and differences
    4. Be familiar with the interfaces for each data structure

4. Sets and Maps
    1. Be familiar with the Set interface and its three concrete implementations
        1. HashSet
        2. LinkedHashSet
        3. TreeSet
    2. Be prepared to compare and contrast the various Set types
    3. Understand the benefits and limitations of Sets compared to other data structures
    4. Be familiar with the Map interface and its three concrete implementations
        1. HashMap
        2. LinkedHashMap
        3. TreeMap
    5. Be prepared to compare and contrast the various Set types
    6. Understand the benefits and limitations of Maps compared to other data structures
5. Developing Efficient Algorithms
    1. Understand what efficiency is and why it is important
    2. Understand Big O Notation
    3. Understand time complexity and space complexity, and their difference
    4. Understand types of algorithm analysis
        1. best-case input/analysis
        2. worst-case input/analysis
        3. average-case analysis
    5. Understand and be ready to match up the Big O measurement with its name, and their order
        - O(1) “Constant time”
        - O(log n) “Logarithmic time”
        - O(n) “Linear time”
        - O(n log n) “Log-Linear time”
        - O(n2) “Quadratic”
        - O(n3) “Cubic”
        - O(2n) “Exponential”
6. Sorting Be familiar with the following algorithms, their implementation, and their time complexity
    1. Insertion Sort
    2. Bubble Sort
    3. Merge Sort
    4. Quick Sort
    5. Heap Sort