

Review Sheet for Exam 1

Exam 1 covers topics from textbook Chapter 6, Chapter 18, Sections 19.1–19.5, and Chapter 20. The topics include container classes and iterators (the collections API), general trees, binary trees, binary search trees (unbalanced, AVL, and red-black trees), and hash tables. Questions will be of four basic types:

- Factual questions that test your knowledge of basic terminology and facts (matching, definitions, or short answer).
- Conceptual questions about data structures that might involve describing in words or pictures some data structure or operation on a data structure.
- Programming questions where you have to write Java code for a data structure implementation (giving data representation and/or methods) or code using a data structure.
- Analysis questions where you have to analyze the efficiency of data structure operations or code that uses the covered data structures.

The following are some sample questions that give a feel for the types and difficulty of questions that you might see on the exam. For each question I have also estimated a reasonable upper bound on the amount of time to answer the question — that information won't be on the exam, but I have put it here so you can evaluate how efficiently you are answering the questions.

1. (Binary Trees/Programming – 15 min) Give Java code that defines a `BinaryNode` class to represent a node in a binary tree (only give the member variables, not any methods). Using your definitions, write a function that counts the number of nodes that have exactly 1 child.
2. (Binary Trees/Analysis – 12 min) Prove that the black height of a red-black tree with n nodes is $O(\log n)$. Don't just write formulas — some explanation of what the formulas represent is required!
3. (Binary Trees/Conceptual/Factual – 5 min) Draw a simple (unbalanced) binary search tree after inserting the following keys (in this order): 181, 632, 439, 906, 108, 589, 398, 732, 836. What is the depth of the resulting tree? How many leaves does it have?

4. (Hash Tables and BSTs/Factual – 3 min) Describe one advantage and one disadvantage of hash tables as compared to red-black trees.
5. (Cumulative/Programming and Analysis – 12 min) Write a function which takes two `Collection` objects and returns boolean indicating whether any item in the first collection appears in the second one. What is the time complexity of your function if the collections are linked lists? What is the time complexity if they are red-black trees?
6. (Hash Tables/Conceptual – 8 min) Consider an 8-bucket hash table using separate chaining — hash function values range from 0 to 7, and consider using a hash function that gives the following values:

Key 632 hashes to 0

Key 906 hashes to 2

Keys 108, 732, and 836 hash to 4

Keys 181 and 589 hash to 5

Key 398 hashes to 6

Key 439 hashes to 7

Draw the hash table after insertions are performed in this order: 181, 632, 439, 906, 108, 589, 398, 732, 836. Could we use an 8-entry hash table using linear probing for this same data? Why or why not?