

COSC-211 Spring 2019 Midterm 2 Topics

This exam will cover all material since the previous exam. It is not cumulative, except in the sense that some of the material from this section of the course builds on material from the first part. The primary goal of this exam is to assess your understanding of the new material since the first midterm.

This is *not* a comprehensive study guide. There may be topics that we have discussed in class or that have come up on homework that are not on this list. You are responsible for all of the course material up to this point, including both in-class material and homework.

1. Java topics

(a) Interfaces

- What it is
- Why we'd use one
- How to write one

2. ADTs and Data Structures

(a) Linked Lists

- What it is
- How to implement one
- Stack/Queue implementation using a Linked List
- Asymptotic analysis of the above

(b) Dictionaries

- What are the supported operations?
- Binary Search Tree implementation
 - Properties
 - How to perform dictionary operations
- Hash Table implementation
 - Chaining
 - Open addressing (linear probing, double hashing)
- Asymptotic analysis of each of the above

(c) Red-Black Trees

- Properties
- Why properties guarantee $O(\lg n)$ worst case height

3. Theoretical topics

(a) Amortized analysis

- What is it?

- Array expansion amortized proof

Some types of questions that I might ask (again, not comprehensive):

- Given a new tree operation that you haven't seen before, analyze its runtime.
- Given a set of data values, show the BST/red-black tree/hash table that results when all of the data is added.
- Write a method to perform a new operation on a BST.
- Compare the runtime of different implementations of the same abstract data type.
- Given a target application, discuss which of a Hash Table, unbalanced BST, or red-black tree would be the best ADT for the application and why.

You've gotten lots of practice, on homework, with implementing the data structures we've discussed. The exam will focus more on using the data structures and understanding their performance.