

COMP 2231_SW3 - Data Structures and Algorithms (Fall 2020 Carruthers)

[Dashboard](#) / [My courses](#) / [COMP_2231_A03_202115](#) / [Sections](#) / [Assessments Overview](#)

/ [Assignment 5: Heaps and Priority Queues, Graphs, & Hashing \(8%\)](#)

Assignment 5: Heaps and Priority Queues, Graphs, & Hashing (8%)

According to our suggested 13 week course schedule, this programming project should be completed and submitted by the Monday of Week 13. It is worth 8% of your final grade. Please refer to the "Assignments Overview" for details on the marking rubric and submission of work.

1. PP 21.5 (page 831)

As described in Section 21.5 of your textbook, it is possible to make the heap sort algorithm more efficient by writing a method that will build a heap in place using the array to be sorted. Implement such a method and rewrite the heap sort algorithm to make use of it.

Show test cases for various inputs, including sorted and unsorted inputs.

2. PP 24.3 (page 902)

Complete the implementation of `Graph.java` using an adjacency matrix that was presented in this chapter. Show test cases for all implemented methods, being sure to test all edge cases where possible.

HINT: Identify **all** of the missing implementations in `Graph.java` before you start. Some methods like `size()` and `getIndex(T vertex)` should be tackled first.

3. PP I.4 (page 1020)

Implement a dynamically resizable hash table to store book names and 10-digit ISBN numbers. Use an initial table size of 11 and a load factor of 0.70. Use the extraction method with division using the last three digits of the ISBN as the first hash address. For a collision, use linear probing. For your testing, make sure you deliberately code ISBNs so that collisions will occur, and demonstrate how the same set of values can result in different hashes, depending on the order in which they are added. Test adding items until the hash table must be resized. Test finding and removing items from the hash table. (Adapted from PP I.4.)

| Criteria | Weighting |
|---|-----------|
| Correctness of solution: Algorithm is implemented and produces correct results for the stated problem. | /4 |
| Testing: Submission of test exhibits to indicate the solution works for a range of cases (e.g., minimum and maximum inputs) and handles unexpected exceptions. | /2 |
| Comments and documentation: Source code contains comments that explain in plain English what the code is intended to do. Note: Javadoc style is not required. | /2 |
| Total | /8 |

Submission status

| | |
|---------------------|--------------------------------|
| Attempt number | This is attempt 1. |
| Submission status | No attempt |
| Grading status | Not graded |
| Last modified | - |
| Submission comments | ▶ Comments (0) |

Add submission

You have not made a submission yet.

[◀ Assignment 4: Trees & Binary Search Trees \(8%\)](#)

Jump to...

[Quiz 1 \(2%\) ▶](#)



805 TRU Way
Kamloops, BC V2C 0C8
Canada
Contact Us

TRU Student Links

- Student Services
- Financial Aid
- Library
- Bookstore
- Student Email
- Self-service Password Portal
- IT Support

Student Moodle Support

- Logging In
- Getting Started

Faculty Moodle Support

[Logging In](#)

[Requesting a Course](#)

[\(Campus Faculty Only\)](#)

[Course Search](#)

[Campus Faculty Support](#)

[OLFM Support](#)

TRU's Kamloops campus is situated on the traditional and unceded lands of the Tk'emlúps te Secwépemc within Secwépemc'ulucw, the traditional territory of the Secwépemc people.

©2021 - THOMPSON RIVERS UNIVERSITY