Bishop's University
Department of Computer Science

CS454
Complements in Data Structures and Algorithms

Fall 2020

Instructor: Dr. Madjid ALLILI
Email: mallili@ubishops.ca
Office: Johnson 125
Phone: (819) 822 9600 2740

Lectures: Online Asynchronous (not scheduled).

Office Hours:

- We will explore Moodle and Teams functionalities to schedule online meetings. More details will be given next week.
- You can also reach me via old communication channels: Email and by calling (if you are local) my office phone number above.

Synopsis. In computer science, data structures and algorithms are extremely important for organizing data in a computer so that it can be accessed and manipulated efficiently to solve a given problem. The study of data structures and algorithms is a fundamental component of computer science education upon which many other computer science fields are built. A basic knowledge of this topic is indispensable for students who wish to do work in design, implementation, testing, or maintenance of any software system.

Course Description. This course is intended to make you familiar with fundamental data structures like stacks, queues, trees, priority queues, hash tables and graphs and with some popular algorithms as techniques for problem solving. First, you learn about algorithms efficiency: how to measure it, and therefore how to improve it. Then, different techniques for writing algorithms are discussed, with examples in each case. Data structures are presented as abstract data types, each with their most important operations. Time complexity and the Θ notation are introduced in order to analyze and compare the efficiency of various algorithms. Programming examples are presented in C++ and therefore a basic knowledge of the language is assumed.

Prerequisites. There are both formal and informal prerequisites to the course. The formal prerequisites are conditional admission to a graduate program with this course as a preparatory co-requisite or CS211-Programming Methodology or equivalent and a basic knowledge of C++. The informal prerequisites are plenty of time, and willingness to experiment and do work. In some respects the informal prerequisites are more important than the formal ones.

Textbook. There is no specific textbook required for this course. Slides and handouts and any other relevant material will be posted on `Moodle`. All the material of the course will be extracted from the following texts that can easily found online (please note that I cannot legally provide copies of these texts):

1. *Data Abstraction & Problem Solving with C++*, M. Carrano et al., 7/e, Pearson, 2016.
2. *Foundations of Algorithms*, R.E. Neapolitan, 5/e, Jones & Bartlett Learning, 2015.
3. *The C++ Programming Language*, B. Stroustrup, 4/e, Addison-Wesley, 2013.
4. *Data Structures & Algorithm in C++*, M.T. Goodrich et al., 2/e, Wiley, 2011.
5. *Introduction to Algorithms*, H. Cormen et al., 3rd Edition, The MIT Press, 2009.
6. *Data Structures & Algorithms in Java*, R. Lafore, 2/e, Sams Publishing, 2002.

Course Contents. Please note that this is a tentative sequence and that nothing is written in stone. If I see the need to spend more time on some difficult concept and develop it more, I shall do so.

1. **Overview:** Overview of data structures and Algorithms.
2. **Algorithms and Complexity:** Worst case, best case, average case, asymptotic time complexity, order notations.
3. **Arrays and Some Sorting Techniques:** Arrays, ordered arrays, bubble sort, selection sort, insertion sort, sorting objects.
4. **Stacks and Queues:** Stacks, queues, parsing arithmetic expressions.
5. **Linked Lists:** Simple linked lists, sorted lists, doubly linked lists.
6. **Recursion Algorithm:** Examples, recursive binary search, divide-and-conquer algorithms.
7. **Divide and Conquer Algorithms and Advanced Sorting:** Mergesort, partitioning, quicksort, complexity analysis.
8. **Binary Trees:** Definitions, binary trees, tree traversal, efficiency of binary trees.
9. **Other Structures and Algorithms:** Hash tables, heaps, priority queues, graphs, greedy algorithms and dynamic programming.

Evaluation. Final grades in this course will be based on performance on (1) Assignments, (2) Quizzes, and (3) Final Exam. Grades in these individual areas will be weighted as follows:

1. **Assignments (programming, homework) (20%)**
2. **Quizzes (30%)**
3. **Final Exam (50%)**

General Notes.

1. Late assignments will not be accepted unless a special permission is obtained from the instructor.

2. There are no supplemental exams.

3. I will give more details at the beginning of the third week about the way I am planning to hold the quizzes and the final exam.

4. Students are strongly advised NOT to make final plans for travel or employment during the final exam period since special arrangements will NOT be made for examinations that may conflict with such plans.

5. Please read the academic calendar regarding academic integrity and attendance. **Strict disciplinary action will be taken in cases of cheating and plagiarism.**

Special Needs. If you have a need for special accommodations, please discuss this with me at the beginning of the semester. As well you must contact the Student Accessibility & Accommodation Services (http://www.ubishops.ca/future-current-students/student-campus-life/student-services/student-accessibility-accommodation-services/) to obtain authorization for any special arrangements.