McGill University

www.cs.mcgill.ca

Course Outline

Course Name: Algorithms and Data Structures
COMP 251 Winter 2025

Instructors: David Becerra

Office Hours: TBD
Email: david.becerra@mcgill.ca
cs251-winter@cs.mcgill.ca

Course Objectives:

Welcome to COMP-251! Please read this document carefully and keep it for reference throughout the term.

COMP-251 refers to the design of correct and efficient algorithms. A correct algorithm must take all possible instances of a described problem to the right result. An efficient algorithm must find the right result in a reasonable amount of time as a function of the size of the input. To train ourselves in the process of designing correct and efficient algorithms, we will perform the following activities:

- To review algorithms' analysis and design techniques. We will also cover and practice on instances where those techniques can be applicable.
    - To review (and analyse) correct and efficient algorithms in the state of the art.
    - To design correct and efficient algorithms that solve 'real' world problems.

COMP-251 will be a combination of theory and practice. Both of them will interact during the term. The course uses the Java programming language, where questions in assignments and exams may require you to write a Java program.
Learning how to design algorithms is not easy; it is not a set of facts that one can simply memorise. You will need to study and practice to master the main concepts. This course aims to teach students a way of thinking that will enable them to build correct and efficient algorithms.

Primary learning objectives:

By the end of this course, students will be able to:

- Analyze the correctness of an algorithm.
- Analyze the efficiency of an algorithm.
- Know algorithms in the state of the art.
- Design correct and efficient algorithms.
- Implement the designed algorithms.

Pre-requisites: *COMP 250 Introduction to Computer Science and MATH 240 Discrete Structures*

Restrictions:

Not open to students who have taken or are taking COMP 252.

We share the same objectives: The description of various computational problems and the algorithms that can be used to solve them, along with their associated data structures. Proving the correctness of algorithms and determining their computational complexity.

Required Software:

- Java https://www.java.com/en/
- myCourses https://mycourses2.mcgill.ca/
- Discussion Board TBD
- Autograder TBD

Class Material:

All the material needed for this class will be available on mycourses. There is no required textbook. However, we recommend the following textbooks from which most lectures will be based upon:

- [CLRS2009] Cormen, Leiserson, Rivest, & Stein, *Introduction to Algorithms*.
- [KT2006] Kleinberg & Tardos, *Algorithm Design*.

Lecture slides will be made available in PDF form on MyCourses. Lectures will be recorded and available on MyCourses. Instructions to borrow an E-book online are available at http://www.mcgill.ca/library/find/ebooks/borrowing-ebooks/.

Course Delivery:

Lectures: Live-In Person (recorded) or pre-recorded (in exceptional circumstances) on Tuesday & Thursday at 11:35 - 12:55pm (EDT).

Assignments: The guidelines will be released in MyCourses and the solutions are expected to be submitted in our autograder. We keep the right to modify the software during the term (in exceptional circumstances).

Contact: All course-related email should be sent to cs251-winter@cs.mcgill.ca. Emails sent to personal addresses will not likely be seen.

Office hours:

Please consult the COMP251 calendar in mycourses for the most up-to-date schedule including office hours and other important dates.

Tentative Grading

Work	Weight	Comment
Quizzes (3)	0%	Quizzes to help you prepare for the exams.
Assignments (3)	27%	Three programming assignments. ● 1 for Data structures (9%). ● 1 for Paradigms (9%). ● 1 for Graph Theory (9%).
Midterm Exams (2)	40%	Midterm 1 [10-Feb : 13-Feb] 20%. ● Topics: Review + Data Structures. Midterm 2 [10-Mar : 13-Mar] 20%. ● Topics: Paradigms.
Final Exam (1)	33%	One final exam to be held during the exam period. ● Main Topic: Graph Theory. ● Secondary Topics: General + DS + Paradigms

Tentative Course Outline

Section	Lecture	Topic	Book/Chapter	Date	Event
General	L1	Introduction - Review I	N/A	07-Jan	
	L2	Review II	Chapter 3 of [CLRS2009]	09-Jan	
	L3	Review III	Chapter 3 of [CLRS2009]	14-Jan	
	L4	Review IV - Hashing I	Chapter 3 of [CLRS2009]	16-Jan	
Data Structures	L5	Hashing II	Chapter 11 of [CLRS2009]	21-Jan	A1 released (20-Jan)
	L6	Disjoint Sets	Chapter 21 of [CLRS2009]	23-Jan	
	L7	Heaps & HeapSort	Chapter 6 of [CLRS2009]	28-Jan	

	L8	BST and AVL trees	Chapter 12 of [CLRS2009]	30-Jan	
	L9	Red Black trees	Chapter 13 of [CLRS2009]	04-Feb	
Design and Analysis	L10	Complete Search	Chapter 13 of [CLRS2009]	06-Feb	A1 Due (8-Feb)
	L11	Divide and Conquer I	Chapter 4 of [CLRS2009]	11-Feb	A2 Released (8-Feb)
	L12	Divide and Conquer II	Chapter 5 of [KT2006]	13-Feb	
		Midterm#1	L1 - L9	TBD	[10-Feb : 13-Feb]
	L13	Dynamic Programming (DP) I	Chapter 6 of [KT2006]	18-Feb	
	L14	Dynamic Programming (DP) II	Chapter 6 of [KT2006]	20-Feb	
	L15	DP III and Greedy I	Chapter 6 of [KT2006]	25-Feb	
	L16	Greedy II	Chapter 16 of [CLRS2009]	27-Feb	A2 Due (01-Mar)
		WINTER BREAK		04-Mar	
				06-Mar	
Graph Theory	L17	Graph Traversals		11-Mar	
	L18	CC / Topo-Sort	Chapter 22 of [CLRS2009]	13-Mar	A3 released
		Midterm#2	L10 - L16	TBD	[10-Mar : 13-Mar]
	L19	Network Flow 1	Chapter 26 of [CLRS2009]	18-Mar	
	L20	Network Flow 2	Chapter 26 of [CLRS2009]	20-Mar	
	L21	Single Source Shortest Path	Chapter 24 of [CLRS2009]	25-Mar	
	L22	Minimum Spanning Trees	Chapter 23 of [CLRS2009]	27-Mar	
	L23	Bipartite Graphs	Chapter 1 of [KT2006]	01-Apr	
General	L24	Amortized Analysis	Chapter 17 of [CLRS2009]	03-Apr	A3 Due
	L25	Randomized Algorithms	Chapter 6 of [KT2006]	08-Apr	
	L26	Probabilistic Algorithms	Chapter 5 of [CLRS2009]	10-Apr	

Learning and Performance Descriptors for programming-based assessments.

Trait	Mastery	Proficient	Developing	Beginning
Correctness	The solution works correctly on all inputs and meets all specifications.	The solution meets most of the specifications. It is incorrect only in a few instances.	The solution is incorrect in many instances.	The solution does not run at all or produces incorrect results almost always.
Algorithm Design	The choice of algorithms, data structures, or implementation techniques is very appropriate to the problem.	The choice of algorithms, data structures, or implementation techniques is mostly appropriate to the problem.	The choice of algorithms, data structures, or implementation techniques is mostly inappropriate to the problem.	Fails to present a coherent algorithm or solution.
Performance	The solution meets all performance expectations	The solution meets most performance expectations	The solution meets few performance expectations	The solution is not performant.
Readability	The solution is well organised according to course expectations and it is very easy to follow without additional context.	The solution is mostly organised according to course expectations and overall easy to follow for someone with context.	The solution is readable only by someone who knows what it is supposed to be doing.	The solution is poorly organised and very difficult to read.
Reusability	The entire solution is composed of reusable units.	Most of the solution is composed of reusable units.	Some parts of the solution are composed of reusable units.	The solution is not organised for reusability.
Documentation	The solution is well documented according to the course expectations.	The solution is well documented according to the course expectations.	The solution documentation lacks relevancy or disagrees with course expectations.	The solution lacks documentation.

Learning and Performance Descriptors for Programming-based assessments and Exams

Trait	Mastery	Proficient	Developing	Beginning
Understanding of the problem	Demonstrates a deep understanding of the problem, identifying all key components and potential challenges.	Shows a good understanding of the problem, identifying most key components and challenges.	Shows a basic understanding of the problem but may miss some key components or misunderstand certain aspects.	Struggles to understand the problem and its requirements, leading to significant inaccuracies in the solution.
Strategy	Articulates a clear strategy to correctly solve the problem, according to their understanding of the task at hand.	Articulates only parts of an appropriate strategy, but most key elements are present.	Articulates a strategy that is only partially useful, key elements might be missing, or inappropriate steps are provided.	Fails to provide a strategy to address the problem.
Readability	The solution is well organised according to course expectations and it is very easy to follow without additional context.	The solution is mostly organised according to course expectations and overall easy to follow for someone with context.	The solution is readable only by someone who knows what it is supposed to be doing.	The solution is poorly organised and very difficult to read.
Algorithm Design	The choice of algorithms, data structures, or implementation techniques is very appropriate to the problem.	The choice of algorithms, data structures, or implementation techniques is mostly appropriate to the problem.	The choice of algorithms, data structures, or implementation techniques is mostly inappropriate to the problem.	Fails to present a coherent algorithm or solution.
Validation	Implements a structured strategy to validate the satisfaction of all the problem's requirements.	Implements a good strategy to validate the satisfaction of some of the problem's requirements.	Implements a weak strategy to validate the satisfaction of few of the problem's requirements.	The validation strategy is weak and incomplete.
Correctness	The solution works correctly on all instances of the problem and meets all specifications.	The solution meets most of the specifications. It is incorrect only in a few instances.	The solution is incorrect in many instances.	The solution produces incorrect results almost always.

General Information

Communication:

- **My Courses**: All official communication, including announcements, lecture material, assignments, grades will be found on My Courses.
- **Course Discussions**: TBD
- **Course Email:** All course-related email should be sent to cs251-winter@cs.mcgill.ca. Emails sent to personal addresses will not likely be seen.
- **Private Email**: The professor and TA have private email accounts that you may 'exceptionally' also use, however these communication channels are for personal queries. For example: if you have a private and very personal message that you want only the instructor to address then you can email the instructor directly.
- **Office Hours**: Please take a look at all posted office hours. Come to those times without appointment.
- **After lecture**: Some optional time will be available just after class to ask questions. I do not guarantee the length of this time since other constraints may interfere.

Assignments & Tests:

- **Assignments Delivery**: All assignments are picked-up from myCourses and submitted to TBD.
- **Late Policy**: You will be notified in advance of assignment due dates. Particularly, due date/time, location/mode for returning your solutions, and accepted formats will be announced in class, stated in the assignment outline and/or indicated on mycourses. Failure to return your assignment in time will result in penalties and possibly absence of grading. **Late submission of 24h or less will receive a penalty of 20%. In all other cases, your assignment will be refused and not graded.**
- **Additional Work**: Students will not be given the opportunity to complete additional work to upgrade their grade.
- **Grading Policy**: Assignments and exams may include guidelines and require particular formatting procedures. **Solutions that do not follow the required format will not be graded.** The quality of the presentation of your solution is important. Unreadable material, cryptic notations, or bad organization of the material may result in absence of grading. Clarity of your explanations will be an integral part of your final grade.
- **Cheating/Collaboration**: Collaboration is encouraged but your discussions should be public in the sense that anyone including the professor should be allowed to listen in. Assignments are original works created by the student alone. You are permitted and encouraged to have conversations with other students concerning the contents of the assignments and how to do them, but your work must be original. If two or more assignments are found to be identical (or portions of assignments) then all parties will lose points. This includes the student who permitted their assignment to be copied. This includes written solutions and software source code.
- **Exam Policy**: Students are responsible for all materials for the tests and exams. Exams will be a combination of all types of questions based on all sources, and students may be required to integrate theoretical concepts from the text to substantiate their arguments. Crib sheets, calculators, dictionaries are not permitted during an exam or test unless specifically stated by the professor.

Additional Information:

The course slides are not meant as a complete set of notes or a substitute for a textbook, but simply constitute the focus of the lecture. Important gaps are left in the slides that are filled in during class, thus **l**ecture attendance should be considered essential.

The material covered in the classroom will be used to supplement textbook readings.

Every chapter should be read twice. The first reading should be done prior to attending class and the second reading should be done after the class discussion of the chapter.

Right to submit in English or French written work that is to be graded.

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Academic Integrity: *Code of Student Conduct*

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/integrity for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples reseignements, veuillez consulter le site www.mcgill.ca/integrity).

Final Exam Policy: *Regulations*

Students should not make other commitments during the final exam period. Vacation plans do not constitute valid grounds for the deferral or the rescheduling of examinations. See the Centre Calendar for the regulations governing Examinations: https://www.mcgill.ca/exams/regulations

Students are required to present their I.D. Card (with photo) for entrance to their examination.

Final Exam Policy: *Conflicts*

If you are unable to write your final examination due to scheduling conflicts, you must submit a Final Exam Conflict Form with supporting documentation at least **one month** before the start of the final examination period. Late submissions will not be accepted. For details, *https://www.mcgill.ca/exams/dates/conflicts*

Final Exam Policy: *Exam Timetable*

Examination schedules are posted at the Centre and on the following page approximately 6-8 weeks before the examination period commences https://www.mcgill.ca/exams/dates

The Centre cannot provide examination dates over the telephone.

Student Rights and Responsibilities:

Regulations and policies governing students at McGill University can be downloaded from the website: https://www.mcgill.ca/students/srr/

Students Services and Resources:

Various services and resources, such as email access, walksafe, library access, etc., are available to McGill students: https://www.mcgill.ca/studentservices/

Various services and resources are offered to computer science students: https://mcgill-csus.ca/

Minerva for Students: http://www.mcgill.ca/minerva-students/

Important Note:

In the event of extraordinary circumstances beyond the University's control, the evaluation scheme in a Course is subject to change, provided that there be timely communications to the students regarding the change.

Land acknowledgement:

McGill University is on land which has long served as a site of meeting and exchange amongst Indigenous peoples, including the Haudenosaunee and Anishinabeg nations. We acknowledge and thank the diverse Indigenous people whose footsteps have marked this territory on which people of the world now gather. Please see here for more details: https://www.mcgill.ca/edu4all/other-equity-resources/traditional-territories .