COMP 5112/COMP4900G: Algorithms for Data Science (Fall 2024 Term)

Weekly Schedule

Instructor: Anil Maheshwari Office: Herzberg Building 5125B E-mail: anil@scs.carleton.ca

Lectures: Lectures on Wednesdays 14:35 - 17:25 AM. See public class schedule/Brightspace for the room location in River Building Ground Floor.

Office hours: Wednesdays 10:15-11:45 AM (HP 5125b). All general announcements will be made during the class and/or via the Brightspace system.

Teaching Assistant: Very unlikely

Course objectives:

To learn some of the algorithmic techniques to handle data science problems. Emphasis is on providing correctness proofs, establishing competitive ratios, and analyzing computational complexity for each of the algorithms discussed during the course.

Topics may include:

- Approximation algorithms design techniques
- Dimensionality Reduction
- Online Algorithms (including the role of Primal-Dual LPs in their analysis)
- Finding Similar Items using Locality-Sensitive Hashing
- Nearest Neighbor Searching
- Clustering
- FPT Algorithmic Design Techniques
- Algorithmic Aspects of Social Networks (Graph Partitioning, Searching various substructures)

These topics may be adjusted based on the background, interests of the students, and the amount of lecture time available.

Required Background:

We will cover a spectrum of techniques from the design and analysis of algorithms. It is assumed that you have an **excellent** grasp on:

- Analysis of algorithms (O-notation, recurrences, and complexity analysis)
- Elementary probability theory including expectation, indicator random variables, Variance, Markov's Inequality (contents of COMP 2804)
- Basic data structures (lists, trees, hashing, BST)
- Discrete mathematics (counting, permutations and combinations, graph theory, proof techniques: induction, contradiction, ..)
- Algorithmic techniques (divide and conquer, greedy, dynamic programming, BFS, DFS, Connectivity, Shortest Paths, and what is NP-Completeness)
- Linear Algebra (Eigenvalues/vectors, Rank-Nullity, Vector Spaces, Norms).

Note that there will not be sufficient time to review the background material to a satisfactory level during the course. (In nutshell you must have a background that is equivalent to the following Carleton Courses: COMP 1805, COMP 2402, COMP 3804, and a course in Linear Algebra.)

Reference Material:

- Foundations of Data Science by Blum, Hopcroft and Kanan.
- <u>My Notes on Topics in Algorithm Design</u>
- Mining of Massive Data Sets: <u>mmds.org</u>
- Several Research Articles that will be mentioned during the course.

Useful references related to various topics. This will get modified as we go along in the course.

- Linear Algebra:
 - Eigenvalues (<u>1</u><u>2</u>)
 - <u>Video Lectures</u> and the <u>book</u> of Gilbert Strang
 - Chapter on Matrices for CS in My Notes on Topics in Algorithm Design
- Probability:
 - STAT110 on Youtube
 - Introduction to Probability book by Blitzstein and Hwang
 - Discrete Structures for Computer Science: Counting, Recursion, and Probability by Michiel Smid
 - Chapter on Probability in My Notes on Topics in Algorithm Design
 - Introduction to R
- Locality Sensitive Hashing
 - Chapter 3 in Mining of Massive Data Sets by Stanford Group (mmds.org)
 - Useful references on LSH
 - STOC98 paper of Indyk-Motwani
 - Chapter 9 on LSH in My Notes on Topics in Algorithm Design
- Data Streaming
 - Mining Data Streams chapter of MMDS
 - Wikipedia Article
 - DGIM Article
 - Count-Min Sketch Article
 - <u>AMS Article</u> (Approximating frequency moments)
 - Knuth's Article CVM Algorithm
 - Flajolet and Martin's Article on number of distinct elements in a stream
 - Some parts are covered in the Chapter in My Notes on Algorithm Design
 - My talk on Bloom Filters and Count-Min-Sketch
- Online Bipartite Matching
 - Karp, Vazirani, and Vazirani, An optimal algorithm for online birpartite matching ACM-STOC 1990
 - Devanur, Jain and Kleinberg, Randomized primal-dual analysis of ranking for bipartite matching, ACM-SIAM SODA 2013.
 - Mehta, Saberi, Vazirani and Vazirani, AdWords and generalized online matching, Jl. ACM Vol. 54, 2007.
 - Kalyanasundaram and Pruhs, An optimal deterministic algorithm for online b-matching, Theoretical Computer Science 233(1-2): 319-325, 2000.
 - Chapter on Advertisement on the Web from the MMDS Book.
 - Eden, Feldman, Fiat, and Segal, An Economics-Based Analysis of RANKING for Online Bipartite Matching, arXiv, 2020.
 - Echenique, Immorlica, and Vazirani, Online and Matching-Based Market Design, Cambridge University Press, 2023.
 - See Section 11.1 11.4 of My Notes
- Randomized Load Balancing & Perfect Hashing
 - http://pages.cs.wisc.edu/~shuchi/courses/787-F09/scribe-notes/lec7.pdf
 - Kleinberg&Tardos Algorithm Design Book, Chapter 13.
- Polynomial Identity Testing
 - DeMillo and Lipton, A probabilistic remark on algebraic program testing, Information Processing Letters 7(4):193-195, 1978.
 - Mulmuley, Vazirani and Vazirani, Matching is as easy as matrix inversion, Combinatorica 7(1):105-113, 1987.

- Mitzenmacher and Upfal, Probability and Computing, Cambridge.
- Motwani and Raghavan, Randomized Algorithm, Cambridge.
- MWIS
 - U. Feige and D. Reichman, Recoverable values for independent sets, Random Structures and Algorithms 46(1): 142-159, 2015.
- Approximation Algorithms using Local Search
 - A. Gupta and K. Tangwongsam, Simpler Analyses of Local Search Algorithms for Facility Location (arXiv Paper)
 - Arya et al., Local search heuristics for k-median and facility location problems, SIAM Jl. Computing 33(3): 544-562, 2004.
 - N.H. Mustafa and S. Ray, Improved results on geometric hitting set problems, Discrete and Computational Geometry 44:883-895, 2010
 - R. Aschner, M.J. Katz, G. Morgenstern and Y. Yuditsky, Approximation schemes for covering and packing, WALCOM, Lecture Notes in Computer Science 7748: 89-100, Springer, 2013.
- More Approximation Algorithms (using LP)
 - Min Cost st-cuts, Multiway Mincuts, Multicuts in Graphs
 - Garg, Vazirani and Yannakakis, Primal-dual approximation algorithms for integral flow and multicuts in trees, Algorithmica 18(1): 3-20, 1997.
 - Garg, Vazirani and Yannakakis, Multiway cuts in node weighted graphs, Journal of Algorithms 50(1): 49-61, 2004.
 - Calinescu, Karloff and Rabani, Approximation algorithms for the 0-extension problem, ACM-SIAM SODA 2001.
 - Fakcharoenphol, Rao and Talwar, A tight bound on approximating arbitrary metrics by tree metrics, JCSS 69(3): 485-487, 2004
 - The design of approximation algorithms, Williamson and Shmoys, Cambridge University Press, 2011.
- Even more Approximation Algorithms
 - Turning down the noise in blogosphere, El-Arini, Veda, Shahaf, Guestrin, KDD 2009.
 - An analysis of approximations for maximizing submodular set function. Mathematical Programming 14, 265-294, 1978.
 - 0
- Multiplicative-Weight Update Method
 - Arora, Hazan and Kale, The multiplicative weights update method: a meta-algorithm and applications, Theory of Computing 8(1): 121-164, 2012.
 - Chapter 11 of my notes.
- Locality-Sensitive Orderings
 - Chan, Har-Peled, Jones, On locality-sensitive orderings and their applications, SIAM Jl. on Computing 49(3): 583-600, 2020.
- Dimensionality Reduction
 - Matousek, Lectures on Discrete Geometry, Volume 212 of Graduate Texts in Mathematics. Springer, New York, 2002.
 - Dubhashi and Panconesi, Concentration of Measure for the Analysis of Randomized Algorithms, Cambridge University Press, 2009.
 - Dasgupta, and Gupta, An elementary proof of a theorem of Johnson and Lindenstrauss" Random Structures & Algorithms, 22 (1): 60-65, 2003.
 - Johnson and Lindenstrauss, Extensions of Lipschitz mappings into a Hilbert space, Contemporary Mathematics 26:189-206, 1984.
 - Chapter 12 of my notes
- Color Coding
 - Alon, Yuster, and Zwick: Color Coding, Jl. ACM 42(4): 844-856, 1995.
 - Articles worth pursuing for projects in addition to above references

0

Grading Scheme: (Tentative)

	COMP 4900 G	COMP 5112
Assignments	30%	30%
End Term Quiz	15%	15%
Project	55%	55%

There are four components:

- End Term Quiz: An End Term Quiz will take place on the last day of classes. It will encompass the course material as well as seminars presented by students.
- Assignments: A couple of assignments during the course. Please only refer to class notes and the reference material listed on the web-page and/or during lectures for solving assignment problems. Please do not collaborate. Please cite all the references used for solving each of the problems. All assignments need to be submitted electronically using the brightspace system.
- Project (COMP5112): (Initial proposal and presentation 10% + Final Report 25% + Final Presentation 20%).

Outline is as follows:

- 1. Pick a topic. (Look at references under "Reference Material" and conferences in related areas. Also, use Google Scholar to see who refers to those papers etc.) You may look for papers/citations in recent proceedings of the ACM-SIAM Symposium on Discrete Algorithms conference and SIGKDD Conference for relevant topics.
- 2. <u>Initial Proposal:</u> Submit one page draft. What is the topic you chose? Why? What problem(s) you will look at? What you plan to do? Outline of sections of your report? Main References. Due (in pdf format) by September

24. Your 5 minute presentation is scheduled on September 25 during the class time slot.

- 3. <u>Final Project Presentation: Scheduled on November 20 & 27</u> during the class. (BTW, we may have to schedule the presentations outside the class time slot.) Presentation is for approximately 15 minutes duration. (End Term Quiz will have questions from these reports/presentations).
- 4. Project Report: Due by November 26. The report format will likely be a research article. Its best to use LaTeX (e.g. see Overleaf). The sections will include:
 - 1. Introduction (Motivation, Problem Statement, Related Work, Short Summary of what you did).
 - 2. Preliminaries (In case you need to discuss some notation, definitions, etc. as background)
 - 3. Main Section How did you solve the problem; State Algorithm; State its Analysis; State its Correctness.
 - 4. Experiments (in case you performed any simulation etc.)
 - 5. Conclusions (Summary + What did you learn? + What do you think can be done in future?)
 - 6. References
 - 7. Report will be approximately 6 pages long and will be posted on the course web-page. Final Exam will have some questions from these reports.
 - 8. You may use a double column format for example the style file from Canadian Conference in Computational Geometry Style File from here: http://vga.usask.ca/cccg2020/CCCG2020-tex-template.zip
 - 9. It will also help the community if you update/create the relevant Wikipedia page relevant to your project. You will be suitably rewarded with bonus marks.
- **Project (COMP 4900):** (initial proposal and presentation 10% + Final Report 25% + Talk 20%)

Outline is as follows:

1. Pick a topic. (Look at references under "Reference Material" and conferences in related areas. Also, use Google Scholar to see who refers to those papers etc.)

people.scs.carleton.ca/~maheshwa/courses/5112/F24/5112-F24.html

You may look for papers/citations in recent proceedings of the ACM-SIAM Symposium on Discrete Algorithms conference and SIGKDD Conference for relevant topics.

2. <u>Initial Proposal:</u> Submit one page draft. What is the topic you chose? Why? What problem(s) you will look at? What you plan to do? Outline of sections of your report? Main References. Due (in pdf format) by September

24. Your 5 minute presentation is scheduled on September 25 during the class time slot.

- <u>Final Project Presentation: Scheduled on November 20 & 27</u> during the class. Presentation is for approximately 12 minutes duration. (End Term Quiz may have questions from these reports/presentations).
- 4. Project Report: Due by November 26. The report format will likely be a research article. Its best to use LaTeX (e.g. see Overleaf). The sections will include:
 - 1. Introduction (Motivation, Problem Statement, Related Work, Short Summary of what you did).
 - 2. Preliminaries (In case you need to discuss some notation, definitions, etc. as background)
 - 3. Main Section How did you solve the problem; State Algorithm; State its Analysis; State its Correctness.
 - 4. Experiments (in case you performed any simulation etc.)
 - 5. Conclusions (Summary + What did you learn? + What do you think can be done in future?)
 - 6. References
 - 7. Report will be approximately 4 pages long and will be posted on the course web-page.
 - 8. You may use a double column format for example the style file from Canadian Conference in Computational Geometry Style File from here: http://vga.usask.ca/cccg2020/CCCG2020-tex-template.zip

SCHEDULE OF FALL 2024 Term

Sep 04: Introduction + MWU Method

- *MWU*
 - Arora, Hazan and Kale, The multiplicative weights update method: a meta-algorithm and applications, Theory of Computing 8(1): 121-164, 2012.
 - Chapter 11 of my notes.

Sep 11:

Sep 18:

Sep 25: Short Presentations on Projects

Oct 02:

Oct 09:

Oct 16:

Oct 30:

Nov 06:

Nov 13:

Nov 20: Presentations on Projects

Nov 27: Presentations on Projects

Dec 04: End Term Quiz

PREVIOUS TERM Schedule:

W1: Introduction + MWU Method (Deterministic Schemes) + LSH (Jaccard Similarity and Signatures)

- *MWU*
 - Arora, Hazan and Kale, The multiplicative weights update method: a meta-algorithm and applications, Theory of Computing 8(1): 121-164, 2012.
 - Chapter 11.1 of my notes.
- LSH
 - Chapter 8.1 and 8.2 in my notes.
 - Useful references on LSH
 - STOC98 paper of Indyk-Motwani

W2: MWU - Randomized Schemes + LSH

- MWU: Chapter 11.2 of my notes
- LSH: Chapter 8.3 and 8.4 of my notes.

W3: LPs using MWU + Sensitive Family of LSH functions + MWIS

- LPs using MWU
 - See Section 11.3 of Notes
 - <u>Summary on MWU</u>
- Sensitive Family
 - See Sections 8.4, 8.5 and 8.6 of Notes
 - Summary on LSH
- MWIS
 - U. Feige and D. Reichman, Recoverable values for independent sets, Random Structures and Algorithms 46(1): 142-159, 2015.
 - Section 14.1.2 of Notes

W4: Short Presentations on Projects + MWIS + Approximation Algorithms - Local Search (Weighted Max-Cut)

- Summary on MWIS
- Local Search
 - A. Gupta and K. Tangwongsam, Simpler Analyses of Local Search Algorithms for Facility Location (<u>arXiv Paper</u>)
 - Arya et al., Local search heuristics for k-median and facility location problems, SIAM Jl. Computing 33(3): 544-562, 2004.
 - N.H. Mustafa and S. Ray, Improved results on geometric hitting set problems, Discrete and Computational Geometry 44:883-895, 2010
 - R. Aschner, M.J. Katz, G. Morgenstern and Y. Yuditsky, Approximation schemes for covering and packing, WALCOM, Lecture Notes in Computer Science 7748: 89-100, Springer, 2013.
 - Chapter 14.2 in Notes
 - <u>Summary on Local Search</u>

W5: Approximation via Local Search (k-median) + Max k-coverage

- Analysis of 5-approximation for k-median in metric graphs.
- Max k-coverage: Maximizing the Spread of Influence through a Social Network, by Kempe, Kleinberg and Tardos.

W6: Locality Sensitive Orderings + Dimensionality Reduction

$people.scs.carleton.ca/\sim maheshwa/courses/5112/F24/5112\text{-}F24.html$

- Chan, Har-Peled, Jones, On locality-sensitive orderings and their applications, SIAM Jl. on Computing 49(3): 583-600, 2020.
 - Quadtrees
 - DFS Traversal
 - Permutations Orderings
- Dimensionality Reduction
 - Matousek, Lectures on Discrete Geometry, Volume 212 of Graduate Texts in Mathematics. Springer, New York, 2002.
 - Dubhashi and Panconesi, Concentration of Measure for the Analysis of Randomized Algorithms, Cambridge University Press, 2009.
 - Dasgupta, and Gupta, An elementary proof of a theorem of Johnson and Lindenstrauss" Random Structures & Algorithms, 22 (1): 60-65, 2003.
 - Johnson and Lindenstrauss, Extensions of Lipschitz mappings into a Hilbert space, Contemporary Mathematics 26:189-206, 1984.
 - Chapter 12 of my notes.

W7: Locality Sensitive Orderings + Dimensionality Reduction

• <u>Summary on Locality-sensitive orderings</u>

W8: Dimensionality Reduction and Online Bipartite Matching

W9: Dimensionality Reduction + Online Matching

- Proof of JL Theorem
- Summary on dimensionality reduction
- LP Duality and Proof of Greedy Matching
- Fractional Matching

W10: Waterlevel Algorithm for Fractional Matching + LP rounding Algorithms

- Summary on Online Bipartite Matching
- Max-Weight Independent Set of Intervals using LP Rounding (see Exercises 14.12-14.20 in Notes)

W11: Approximation Algorithms (using LP)

- Min Cost st-cuts, Multiway Mincuts, Multicuts in Graphs
- Garg, Vazirani and Yannakakis, Primal-dual approximation algorithms for integral flow and multicuts in trees, Algorithmica 18(1): 3-20, 1997.
- Garg, Vazirani and Yannakakis, Multiway cuts in node weighted graphs, Journal of Algorithms 50(1): 49-61, 2004.
- Calinescu, Karloff and Rabani, Approximation algorithms for the 0-extension problem, ACM-SIAM SODA 2001.
- Fakcharoenphol, Rao and Talwar, A tight bound on approximating arbitrary metrics by tree metrics, JCSS 69(3): 485-487, 2004
- The design of approximation algorithms, Williamson and Shmoys, Cambridge University Press, 2011.
- Chapter 14.3 of Notes
- <u>Summary on Approximation Algorithms using Metric LPs</u>

W12: Final Project Presentations + Review

- Additional Material: <u>k++ Means Clustering</u>
- **Dec 10: Final Exam:** 9 11:30 AM. Scheduled by exam services Please crosscheck the room #. Tentatively in TB 219 Expect about 6-8 problems. Please bring a calculator.

Undergraduate & Graduate Academic Advisor:

The Undergraduate Advisor for the School of Computer Science is available in Room 5302 HP; or by email at scs.ug.advisor@scs.carleton.ca. The undergraduate advisor can assist with information about prerequisites and preclusions, course substitutions/equivalencies, understanding your academic audit and the remaining requirements for graduation. The undergraduate advisor will also refer students to appropriate resources such as the Science Student Success Centre, Learning Support Services and Writing Tutorial Services.

Similarly, we have Graduate Advisors. Depending on your program of study and the university, you may contact the relevant graduate advisors.

University Policies

We follow all the rules and regulations set by Carleton University, Dean of Science, and the School of Computer Science regarding accommodating students with any kind of need(s). Please consult with the appropriate authorities to see how you can be accommodated and please follow their procedures. For information about Carleton's academic year, including registration and withdrawal dates, see Carleton's Academic Calendar. Following is a standard list of recommendations that we have been advised to provide you with respect to whom to contact for the appropriate action(s). It is possible that some of these are out of date, please consult the latest recommendations from the Science Faculty.

Pregnancy Obligation. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, visit Equity Services.

Religious Obligation. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, visit Equity Services.

Academic Accommodations for Students with Disabilities If you have a documented disability requiring academic accommodations in this course, please contact the Paul Menton Centre for Students with Disabilities (PMC) at 613-520-6608 or pmc@carleton.ca for a formal evaluation or contact your PMC coordinator to send your instructor your Letter of Accommodation at the beginning of the term. You must also contact the PMC no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). After requesting accommodation from PMC, meet with your instructor as soon as possible to ensure accommodation arrangements are made. For more details, visit the Paul Menton Centre website.

Survivors of Sexual Violence. As a community, Carleton University is committed to maintaining a positive learning, working and living environment where sexual violence will not be tolerated, and survivors are supported through academic accommodations as per Carleton's Sexual Violence Policy. For more information about the services available at the university and to obtain information about sexual violence and/or support, visit: carleton.ca/sexual-violence-support.

Accommodation for Student Activities. Carleton University recognizes the substantial benefits, both to the individual student and for the university, that result from a student participating in activities beyond the classroom experience. Reasonable accommodation must be provided to students who compete or perform at the national or international level. Please contact your instructor with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details, see the policy.

Student Academic Integrity Policy. Every student should be familiar with the Carleton University student academic integrity policy. A student found in violation of academic integrity standards may be awarded penalties which range from a reprimand to receiving a grade of F in the course or even being expelled from the program or University. Examples of punishable offences include: plagiarism and unauthorized co-operation or collaboration.

Plagiarism. As defined by Carleton University Senate, "plagiarism is presenting, whether intentional or not, the ideas, expression of ideas or work of others as one's own". Such reported offences will be reviewed by the office of the Dean of Science. More information and standard sanction guidelines can be found here: <u>https://science.carleton.ca/students/academic-integrity/</u>.

Unauthorized Co-operation or Collaboration. Senate policy states that "to ensure fairness and equity in assessment of term work, students shall not co-operate or collaborate in the completion of an academic assignment, in whole or in part, when the instructor has indicated that the assignment is to be completed on an individual basis".

For this course, the following also holds:

- Students are **NOT** allowed to collaborate on assignments. Please avoid using search engines to look for answers etc. Just a word of caution in theoretically oriented courses, it is important to come up with your own ideas for the proof/an algorithm/a contradiction/etc. Sometimes these are like logical puzzles if somebody tells you a solution then they are trivial and hard part is to come up with a solution. What we want to learn is how to solve them ourselves.
- Past experience has shown conclusively that those who do not put adequate effort into the assignments do not learn the material and have a probability near 1 of doing poorly on the exams/tests.
- · Penalties for academic offences can be found on the ODS webpage: https://science.carleton.ca/academic-integrity/.

Important Considerations:

people.scs.carleton.ca/~maheshwa/courses/5112/F24/5112-F24.html

Late assignments are not accepted. Assignments submissions are handled electronically using the Brightspace system and there is no "grace period" with respect to a deadline. Technical problems do not exempt you from this requirement. You are advised to:

- periodically upload your progress (e.g. upload your progress at least daily)
- attempt to submit your final submission at least one hour in advance of the due date and time.