Course Information
Instructor: Shahin Kamali
Lectures: 10:00 - 11:15 am, Tuesdays and Thursdays in EITC E2 Room: 164
Office: E2-586
Office hours: 11:30 am-12:30 pm on Mondays and 2:00 pm - 3:00 pm on Tuesdays in E2 586 or by appointment
Email: shahin.kamali@umanitoba.ca (allow 48 hours for response)
Piazza: https://piazza.com/umanitoba.ca/fall2019/comp7720

Course Overview. COMP 4060/7720 (Online Algorithms) provides an introduction to online algorithms and their applications. In contrast to the traditional ‘offline’ algorithms, an online algorithm does not have access to the whole input. Instead, it receives its input piece by piece as a sequence. Upon receiving a piece of the input, the algorithm has to take an irreversible decision to process it. Online algorithms have many applications in practice, and have been studied extensively in theory also. Our goal in this course is to design and analyze online algorithms with provable theoretical guarantee on the quality of the resulting solutions. We cover a variety of online problems in different contexts which include system problems (e.g., paging), data structures (e.g., self-adjusting lists), graph problems (e.g., coloring and matching), packing problems (bin packing and scheduling), computational geometry (searching and exploring) problems. We also study alternative analysis techniques which have been recently introduced (e.g., advice complexity and bijective analysis).

Course Goals and Intended Learning Outcomes. This course exposes students to a growing algorithmic field within computer science that has applications in Computer Systems (e.g., caching and resource allocation), data compression (e.g., bZip2 compression scheme), social networks (e.g., graph partitioning). Students will discuss fundamental concepts in design of online algorithms, techniques used to analyze online algorithms in the worst-case scenarios, and important recent related research developments in ‘semi-online’ algorithms where partial information is available about the input sequence. The students are expected to learn to model online problems in their own domain of research, design practical online algorithms, and analyze their algorithms in the framework of competitive analysis (that is, providing worst-case guarantees).

Prerequisites. Students are expected to have a strong background in theoretical computer science, and be familiar with topics in design and analysis of algorithms, data structures, and discrete mathematics, including searching, sorting, asymptotic notations, time complexity analysis, fundamental data structures, and basic concepts in graph theory.
Textbook. A list of required readings will be provided on the course webpage. No book is required to be purchased. The following book is suggested as a reference:

- Borodin and El-Yaniv, Online Computation and Competitive Analysis (2005)

Syllabus. Topics to be covered in the course will include the following, subject to change at the discretion of the instructor and/or based on the learning needs of students.

- An introduction to online computation and competitive analysis
  - The sky-rental problem
- Doubling technique for design of online problems
  - The lost-cow problem
  - Online bidding
- Potential function method and randomization
  - The paging problem, Sleator-Tarjan result
  - Randomized paging
- Self-adjusting data structures
  - List update problem, data compression
  - Self-adjusting binary trees, and dynamic optimality conjecture
- Weighting technique
  - Bin packing
  - Renting servers in the cloud
  - Scheduling
- Graph problems
  - k-server problem
  - work-function family of algorithms
  - Graph coloring
  - Bipartite matching
- Computational geometry
  - Robot searching
  - 2-dimensional bin packing
- Advice complexity of online algorithms
  - Advice complexity of paging, bin packing, list update
- Alternative analysis methods
  - Bijective analysis
  - Relative worst-order analysis

Grading. All students will be required to complete four assignments, a final exam, and a course project (consisting of a project proposal, a class presentation, and a written report). Grades will be calculated according to the following table:

- Assignments: 30 percent.
- Exam 1: 15 percent (closed-book exam).
- Course project: 30 percent.
• Class participation: 10 percent.

<table>
<thead>
<tr>
<th>letter grade</th>
<th>percent grade</th>
<th>grade point</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>90-100</td>
<td>4.5</td>
</tr>
<tr>
<td>A</td>
<td>80-89</td>
<td>4.0</td>
</tr>
<tr>
<td>B+</td>
<td>75-79</td>
<td>3.5</td>
</tr>
<tr>
<td>B</td>
<td>70-74</td>
<td>3.0</td>
</tr>
<tr>
<td>C+</td>
<td>65-59</td>
<td>2.5</td>
</tr>
<tr>
<td>C</td>
<td>60-64</td>
<td>2.0</td>
</tr>
<tr>
<td>D</td>
<td>50-59</td>
<td>1.0</td>
</tr>
<tr>
<td>F</td>
<td>0-49</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Assignments. Assignments will consist of problem sets, seeking constructive solutions to algorithmic problems related to lecture material and assigned reading. Solutions should include sufficiently detailed descriptions, presented clearly and unambiguously. Assignments will be distributed in class during the term. Students will have roughly one week to complete each assignment individually. Solutions must be submitted by the start of class on the due date. We will use [https://crowdmark.com/](https://crowdmark.com/) for assignment submission and marking. Details to be followed. You should include the names of people with whom you have discussed your assignment solution. Cite any sources to which you refer, as you should do when presenting any scientific document. To permit the prompt distribution of solutions and return of marked assignments, late assignments will not be accepted.

Course Project. The purpose of the course project is for students to select and explore a topic related to online algorithms, to study a current research problem in that topic, to make a new contribution on that topic (see below), and to present the results, in both a written report and a class presentation. The nature of the project can vary; examples include:

• Exploring possible solutions to an open problem on a given topic in online algorithms.

• Writing a survey paper on a current topic related to online algorithms

• Writing code to implement and compare the performance of online algorithms for solving a problem; this might involve implementing the offline algorithms for comparison.

As part of the project, students will be asked to submit a preliminary project proposal, a final written report, and to give a class presentation. Students may choose to complete the project individually or in groups of two.
Important Dates (tentative).

- September 5: the first class
- September 27: project proposal due
- October 1: assignment 1 due
- October 14: assignment 2 due
- October 18: chocolate cupcake day
- October 24: exam 1 (in class)
- October 28: assignment 3 due
- November 12-15 fall break
- November 18: VW deadline
- November 21: assignment 4 due
- November 25 - 28: project presentation dates
- December 5: exam 2 (in class)
- December 13: project final report due

Academic Integrity. The Faculty of Science takes academic integrity very seriously. Any evidence of academic dishonesty on assignments, labs and/or tests will be forwarded to the appropriate authorities for potential disciplinary actions. The University Student Discipline By-Law may be accessed at: http://umanitoba.ca/admin/governance/governing_documents/students/student_discipline.html. Information from the Faculty of Science regarding Cheating and Plagiarism can be found at http://umanitoba.ca/faculties/science/undergrad/resources/webdisciplinedocuments.html. Students are encouraged to discuss course concepts and the general interpretation of homework problems with other students in the class. No written record should be taken from such discussion. Each student must work on the final solution of assignment problems independently. On a cover page, each student must list the names of people with whom he or she has discussed the assignment solution. Following conventions for citing reference materials in scientific writing is mandatory. Submitting the work of another person as your own constitutes academic misconduct. Any course work that does not follow these guidelines will be considered plagiarism and will be reported to the Faculty of Science. Students are to abide by the university’s policies regarding academic dishonesty which can be found on this web site: http://umanitoba.ca/student/resource/student_advocacy/academicintegrity/students/.

Class Communication. All announcements for the class will be posted on the webpage of the course as well as on the course Piazza page. The piazza page will also be the central place for class discussions and for any questions about the lectures and assignments.

Please note that all communication between the instructor and you as a student must comply with the electronic communication with student policy (http://umanitoba.ca/admin/governance/governing_documents/community/electronic_communication_with_students_policy.html). You are required to obtain and use your U of M email account for all communication between yourself and the university.

Updated on September 20, 2019.