THE UNIVERSITY OF WINNIPEG

APPLIED COMPUTER SCIENCE

Graduate Course No: ACS-7101/3
Graduate Course Title: ADVANCED DATA STRUCTURES AND ALGORITHMS FOR APPLIED COMPUTER SCIENCE

Instructor Information

Instructor : Dr. Yangjun Chen **Office: 3D27**
E-mail: y.chen@uwinnipeg.ca
Office Hours: 4:00 pm - 5:00 pm on Mon. and Wed.
10:00 am – 5:00 pm Friday
Class Meeting Time: Mon. and Wed. 10:00 am - 11:15 am **Room No:** 3D03
Instructor home page: http://www.uwinnipeg.ca/~ychen2

Important Dates

1. First Class Date: Jan. 08, 2018
2. First Lab Date: N/A
3. Midterm Exam/Tests/Quizzes: Feb. 28, 2018, 10:00 am - 11:15 am
4. Final Exam (Comprehensive): The final examination may be replaced by a project, for which the students are required to implement some challenging algorithms for tree-pattern query evaluation and indexing mechanism (using a computer language) and make tests and comparison.
5. Final Withdrawal Date w/o academic penalty: March 14, 2018
   (A minimum of 20% of the work on which the final grade is based will be evaluated and available to the student before the voluntary withdrawal date)
6. Reading Week: Feb, 18 – 24, 2018 (No classes)
7. The University of closed on February 19th (Louis Riel Day) and March 30th (Good Friday)
8. Other Deadlines (e.g., assignments, term papers/projects):
   three assignments, the deadlines will be determined during the course.

   All assignments are handed in at class on the due date.
   All works must be prepared using a word processor and placed in a folder.
   Late assignments are accepted (up to 1 day late) and receive a 25% penalty.

Course Objectives/Learning Outcomes

In this course, students will study methods for designing efficient data structures and algorithms such as binary search trees, red-black trees, priority queues, minimum spanning trees, strongly connected components, maximum flows, string matching and tree matching, as well as bipartite graphs. Through the study of these data structures and algorithms, students will develop skills to solve hard problems in specialized databases such as Web and Document, DNA and Deductive Databases.

Evaluation Criteria

Assignments (24%)

- Number of Assignments: 3
- Information about assignments
  - *Due Date for each assignment is regularly two weeks later.*
  - *Late work will receive a 25% penalty.*
  - *Work should be typed and handed in through e-mail.*
  - *Work should be prepared in English.*

Midterm Exam/Tests/ Quizzes (26%)

Midterm Exam/Tests/Quizzes: Feb. 27, 2013, 2:30-3:45

Final Exam (50%)

The final examination may be replaced by a project, for which the students are required to implement some challenging algorithms (using a computer language) and make tests and comparison.

Final Letter Grade Assignment

Historically, numerical percentages have been converted to letter grades using the following scale. However, instructors can deviate from these values based on pedagogical nuances of a particular class, and final grades are subject to approval by the Department Review Committee.

A+	90+ - 100%	B	70 - 74%	F	below 50%
A	85 - 90%	C+	65 - 69%		
A-	80 - 84%	C	60 - 64%		
B+	75 - 79%	D	50 - 59%		

Exam Requirements

- *Photo ID is not required.*
- *The book is closed. Calculators/electronic translators can be used.*

Students with documented disabilities, temporary or chronic medical conditions, requiring academic accommodations for tests/exams (e.g., private space) or during lectures/laboratories (e.g., note-takers) are encouraged to contact Accessibility Services (AS) at 786-9771 or accessibilityservices@uwinnipeg.ca to discuss appropriate options. All information about a student's disability or medical condition remains confidential. http://www.uwinnipeg.ca/accessibility.

All students, faculty and staff have the right to participate, learn, and work in an environment that is free of harassment and discrimination. The UW Respectful Working and Learning Environment Policy may be found online at www.uwinnipeg.ca/respect.

Required Text Book(s)/Reading List

The course uses a book:

1. Introduction to Algorithms," 2nd or 3rd edition, by Cormen, Leiserson, Rivest & Stein, The MIT Press, London, 2007. (ISBN: 0-07-297054-5)

Reference book:

2. Aho, A.V., Hopcroft, J.E. and Ullman, J.D., *The Design and Analysis of Computer Algorithms*, Addison-Wesley Publishing Com., London, 1969.

Prerequisite Information* (This information can be found in the UW General calendar)

• Consent of the Department Graduate Program Committee Chair or Instructor.

*Make sure that you have the necessary prerequisites to take this course. If you have not successfully completed the above listed course(s), it is in your interest to drop the course.

Misuse of Computer Facilities, Plagiarism, and Cheating

Academic dishonesty is a very serious offense and will be dealt with in accordance with the University’s policies. Be sure that you have read and understood **Regulations & Policies** #8 in the *2017-2018*UW Course Calendar.

Topics to be covered

1. Algorithm basics
   1.1 Review of basic data structures
   1.2 Mathematical techniques for the analysis of algorithms
2. Algorithm for sorting
   2.1 Merge-sort, correctness proof, and performance analysis
   2.2 Quick-sort, correctness proof, and performance analysis
   2.3 Heap and heap-sort
3. Binary search trees and Red-Black trees
   3.1 Binary trees: querying, insertion and deletion
   3.2 Red-Black trees: insertion and deletion
4. Dynamic programming
   4.1 Assembly-line scheduling
   4.2 Matrix-chain multiplication
   4.3 Elements of dynamic programming
   4.4 Longest common subsequence
5. Greedy algorithms
   5.1 An activity-selection problem
   5.2 Elements of greedy strategy
   5.3 Minimum spanning trees
6. Graph algorithms
   61 Elementary graph algorithms
   6.2 Topological sort

6.3 Strongly connected components
7. Single-source shortest paths
   7.1 The Bellman-Ford algorithm
   7.2 Single-source shortest paths in directed acyclic graphs
   7.3 Dijkstra’s algorithm
8. Maximum flow
   8.1 Flow networks
   8.2 The Ford-Fulkerson method
9. String matching
   9.1 Naïve algorithm for string matching
   9.2 The Knuth-Morris-Pratt algorithm
10. Bipartite graphs
   Lecture notes

Note: all topics listed on the outline may not be covered.

Projects:
(1) Implementing different strategies for query evaluation in XML document databases
(2) Implementing Hopcroft-Karp algorithm for maximum bipartite matching
More projects will be announced.

Guidance to project reports:

1. Introduction (including the problem description, motivation – its significance and application in the computer engineering and industry)
2. Related work (describe some important techniques related to the problem to be addressed)
3. Main thrust (detailed description of the method, formal algorithm, analysis of computational complexities: time and space overhead)
4. Future work (discussion on the possible improvements, or possible extension)
5. Experiments (main data structures used for implementation, description of the data used for tests, test results: charts, histogram, or tables)
6. References