SCICOMP302, Algorithms and Data Structures
[Fall 2021]

algorithms & data structures

[SCICOMP302; Algorithms and Data Structures]

[Fall 2021]

Classroom no:
Class times: **MON 08:45-10:45 THU 13:45-15:45**

Instructor: **Dr. Andrew Brooks**
Email: **a.brooks@ucr.nl**
Biographical details: **https://orcid.org/0000-0003-2414-6322**
Tel: **0118 655 534**
Office no. & location: **Eleanor 1.09**

Office hours: **appointments - see Moodle scheduler activity**

I. Track information

a) Prerequisites for this course: a C grade or better on SCICOMP102.
b) This course is not a prerequisite for other courses.

This course is part of the Computer Science track. For further information about the track, please see the track document available on the UCR intranet or in Moodle.

II. Course description

This course aims to provide the student with knowledge, skills and critical thinking ability in algorithm design and analysis. Inappropriate choice of algorithm and associated data structure can seriously impact on the performance of an application. The study of algorithm design and analysis provides techniques which help minimize the execution time of an algorithm. An emphasis is on the experimental performance analysis of algorithms. The second meeting of the class each week is entirely devoted to laboratory work where students tackle exercises and demonstrate their work. Larger algorithm projects are also undertaken.

There is a written midterm exam and a written final exam. By the end of the course a student will have obtained an understanding of how algorithms have helped shape the modern world.

III. Study Load

This course earns students four credits (equivalent to 7.5 ECTS). The class meets twice a week for two hours. Preparation time is approximately 10 hours per week.

IV. Course materials

The following textbook is required for this course:
Real-World Algorithms: A Beginner's Guide
Panos Louridas

The MIT Press, 2017
ISBN: 9780262035705 (Hardcover)
https://mitpress.mit.edu/books/real-world-algorithms
Real World Algorithms Book Chapters (Jupyter notebooks)
https://louridas.github.io/rwa/pages/book-chapters.html

Other materials will be made available (see Appendix A).

V. Course organization and requirements

The first session of each week will cover content. This session will involve a mix of lecturing, discussion, and various in-class activities (e.g. short presentations by students, online quizzes, one-to-one or group face-offs). Discussion-based learning will involve rotations of discussion group membership.

The second session of each week will be a laboratory in which exercises are undertaken. The instructor will provide assistance where needed and sign out completed exercises when they have been successfully demonstrated.

Homework and laboratories will be issued approximately on a weekly basis. If the number of issued homework or laboratories exceeds 10, the additional work will count as extra-credit. Other extra-credit opportunities may be available at the instructor's discretion.

At least two class moments (of 1 or 2 hours duration) will be provided to allow students the opportunity to work in-class on projects or on revision for written exams and to ask questions of the instructor.

A minimum of two office hours will be offered weekly with the opportunity for students to schedule a 15 minute appointment.

Students are expected to:

- (i) bring the required textbook to class
- (ii) bring their own fully-charged laptops to class
- (iii) read relevant book chapter(s)
- (iv) participate actively in class when asked to do so
- (v) monitor Moodle, the official mode of communication, on a daily basis
- (vi) attend class and inform the instructor beforehand if they cannot attend because of illness or some other urgent reason
- (vii) work consistently on the laboratory exercises outside of class if necessary
- (viii) tackle all assessment individually unless the task is explicitly described as group-based or the instructor gives explicit guidance on acceptable collaborations
- (ix) be able to explain the programming code they have written or made use of
- (x) not use mobile devices in class other than their own laptops for class work
- (xi) not redistribute materials made available in Moodle to third parties
- (xii) attend office hours for help and guidance on any aspect of the course when required

Homework deadlines are firm. Homework should be returned one week from the date of issue. Laboratories have full value for one week from the date of issue. After one week, their value is halved. After two weeks, they no longer have value.

This course is subject to UCR academic rules and procedures. Both students and instructors are required to know and follow these rules and procedures. Students should not commit acts of **plagiarism** (e.g. submitting another person's work) or **collusion** (e.g. teamworking an individual assessment). Students are advised that if they miss more than 6 class sessions they will receive an automatic F (**i.e. 6 absences only are permissible**). Two hours of lateness in attending class meetings will count as one absence.

VI. Assessment

assessed component	**value**	
written midterm	20%	sample questions will be provided
written final	20%	sample questions will be provided
algorithms project 1	10%	
algorithms project 2	10%	*investigative, possibly group-based*
algorithms project 3	10%	*investigative, possibly group-based*
homework (10 each at 1%)	10%	
laboratories (10 each at 2%)	20%	

Homework, laboratories, written midterm and written final will be assessed on correctness of answers. Partial credit will be awarded for partial correctness. Comprehension questions will be asked of laboratory work. Partial credit will be awarded when comprehension questions are not fully answered.

The written midterm and written final may incorporate wild card questions on all course aspects (up to 2% of the 20%).

Projects are assessed on a sliding scale of accomplishment. **For example**,

Grade	Expectations
A	In addition to the expectations under B, results of an experimental investigation which involve code improvement changes are reported and demonstrated.
B	In addition to the expectations under C, results of an experimental investigation into the algorithm(s) and data structure(s) are reported and demonstrated.
C	In addition to the expectations under D, discussed algorithm(s) and data structure(s) are actually those present in the software chosen for investigation.
D	A presentation only is provided which is substantial and addresses important issues.
F	A presentation only is provided which is lightweight and fails to address important issues

Full project specifications are available in Moodle.

VII. Course schedule
The course schedule may be subject to change. For example, if things are progressing very well, it might be possible to occasionally tackle more material in a week. National holidays or special college events may result in some content being covered in less depth. Key dates within any week will be posted in Moodle.

Time	Topics to be discussed	Course material used	Assignments and assessment
Week 1 [30 Aug]	components of performance, describing algorithms, algorithm analysis, experimental algorithmics	see Moodle	homework & laboratory
Week 2 [6 Sept]	reporting computational experiments, the maximum contiguous subsequence sum problem, logarithms and binary search	see Moodle	homework & laboratory
Week 3 [13 Sept]	Random Number Generator (RNG), testing a (P)RNG, classifying sorts	see Moodle Chapter 16	homework & laboratory
Week 4 [20 Sept]	Big-Oh, Big-Omega, Big-Theta, Little-Oh recurrence relations taxonomies of sorting algorithms	see Moodle Chapter 1	homework & laboratory
Week 5 [27 Sept]	Bubble Sort Selection Sort algorithms @ The Java Tutorials	see Moodle Chapter 12	homework & laboratory **project 1 deadline (10%)**
Week 6 [4 Oct]	(two-way) Merge Sort Quicksort	see Moodle Chapter 12	homework & laboratory
Week 7 [11 Oct]	basics of graphs, adjacency-matrix & adjacency-list representations	see Moodle Chapter 2	homework & laboratory
Week 8 [18 Oct]			**midterm exam (20%)**
BREAK [25 Oct]			

Time	Topics to be discussed	Course material used	Assignments and assessment
Week 9 [1 Nov]	unweighted breadth-first search & Dijkstra's algorithm	see Moodle Chapter 7	homework & laboratory
Week 10 [8 Nov]	Bellman-Ford algorithm & topological sort algorithm	see Moodle Chapters 6 & 8	homework & laboratory **project 2 deadline (10%)**
Week 11 [15 Nov]	Kruskal´s algorithm & Prim´s algorithm	see Moodle	homework & laboratory
Week 12 [22 Nov]	search trees & binary search trees & red-black trees	see Moodle	homework & laboratory

Week 13 [29 Nov]	complexity classes, overview of showing problems to be NP-complete	see Moodle	homework & laboratory
Week 14 [6 Dec]	topics such as cryptography & string matching algorithms	see Moodle Chapters 4, 5, & 15	homework & laboratory
Week 15 [13 Dec]		see Moodle	**final exam (20%)** **project 3 deadline (10%)**

VIII. Student learning outcomes

Upon successfully completing this course, a student should be able to:

SLO 1 to estimate and measure the time complexity of an algorithm, expressed using asymptotic notation e.g. Big-Oh

SLO 2 to design, execute, and report on an experimental investigation of algorithm performance

SLO 3 to recite and trace the execution of several sorting and searching algorithms e.g. insertion sort, merge sort, quicksort, binary search on binary search trees

SLO 4 to recite and trace the execution of several graph algorithms e.g. breadth-first search, depth-first search, minimum spanning trees, single-source shortest paths

SLO 5 to investigate and report on the algorithms and their associated data structures used in applications

SLO 6 to demonstrate a knowledge and understanding of two or more advanced topics e.g. cryptography, string matching algorithms, NP-completeness

Period	**Teaching activities**	**Student is able to do**
Weeks 1-6	lectures & homework & laboratories & written midterm examination in Week 8	SLO 1 SLO 3
Weeks 1-5	Project 1	SLO 2
Weeks 6-10	Project 2	SLO 5
Weeks 7, 9-12	lectures & homework & laboratories & written final examination in Week 15	SLO 4
Weeks 13-14	lectures & homework & laboratories & written final examination in Week 15	SLO 6
Weeks 11-15	Project 3	SLO 5

Appendix A Course Materials
Course materials are provided in Moodle.

For example, below is a copy of just some of the materials (the section on algorithm animations).

- Algorithms and Data Structures Animations for the Liang Books

  includes Brute-Force, Boyer-Moore, Knuth-Morris-Pratt, Compute KMP failure function
- Algorithm Animations and Visualizations
- Algorithm Visualizer
- ALGOVIS.IO
- AlViE is a post-mortem algorithm visualization Java environment
- Boyer-Moore Exact Pattern Match
- Data Structure Visualizations (David Galles)
- Data Structure Visualizations (Yves Lucet)

  includes Boyer-Moore & Knuth-Morris-Pratt
- DDS — Dynamic Data Structures Homepage
- galant at GitHub (Graph Algorithm Animation Tool)
  - Algorithm Animation with Galant (2017)
- OpenDSA Data Structures topics for a CS2-level course
- OpenDSA A post-CS2-level course on Data Structures
- Sorting Algorithms Animations (Toptal)
- Vamonos
- VISUALGO
- Wolfram Algorithms DEMONSTRATIONS