Boston University Global Programs
Study Abroad Sydney

BOSTON
UNIVERSITY

CAS CS330 Introduction to Analysis of Algorithms

Syllabus (Sydney/Australia)

Instructor Name: Dr Mahendra Samarawickrama
Contact Information: msamara@bu.edu
Contact Hours: 40

Course Dates: Input course dates
Course Credits: 4 BU credits

Course Description

This subject explores and analyses mathematical modelling of computational problems so that students become adept at investigating and recommending data structures and abstract data types. This encompasses a curriculum on algorithms, algorithmic paradigms, and data structures. This serves to underscore the relationship between algorithms and programming while introducing performance measures and analysis techniques of importance to modern organizations, leaders, and IT professionals. When students complete this subject, they will be equipped with the problem-solving capabilities required for complex business problems.

Course Learning Outcomes

Upon successful completion of this subject, students should be able to demonstrate achievement of the following learning.

1	Investigate data structures and abstract data types.
2	Analyse the programming of data structures and the implementation of abstract data types.
3	Demonstrate how to estimate algorithmic complexity.
4	Recommend data structures and abstract types as solutions for business problems.

Hub Learning Outcomes

BU Hub Units: This course fulfills a single unit in each of Quantitative Reasoning II. and Critical Thinking.

Quantitative Reasoning II:

Learning Outcome 1: Students will frame and solve complex problems using quantitative tools, **such as** analytical, statistical, or computational methods.

This is the *central* course in computer science in which problem solving methods and tools are considered in detail, after students have had 3-5 programming and theory classes.

Learning Outcome 2: Students will apply quantitative tools in diverse settings to answer discipline-specific questions **or** to engage societal questions and debates.

Learning Outcome 3: Students will formulate, and test an argument by marshaling and analyzing quantitative evidence.

Students will marshal arguments and evidence of an empirical nature to test the properties of algorithms.

Learning Outcome 4: Students will communicate quantitative information symbolically, visually, numerically, **or** verbally.

The course gives ample practice in communicating symbolically. Learning Outcome 5: Students will recognize and articulate the capacity and limitations of quantitative methods and the risks of using them improperly.

This course considers the notion of limitations in great detail, including the fundamental limitation of NP-Hard problems!

Critical Thinking:

Learning Outcome 1: Students will both gain critical thinking skills and be able to specify the components of critical thinking appropriate to a discipline or family of disciplines. These may include habits of distinguishing deductive from inductive modes of inference, methods of adjudicating disputes, recognizing common logical fallacies and cognitive biases, translating ordinary language into formal argument, distinguishing empirical claims about matters of fact from normative or evaluative judgments, and/or recognizing the ways in which emotional responses or cultural assumptions can affect reasoning processes.

This course is in many ways the intellectual "center" of the CS major, where a large variety of classic modes of thinking are explored, and their properties exploited to understand a variety of classic algorithms. Proof methods and empirical techniques are used to prevent cognitive bias and logical fallacies, and to rigorously test assumptions.

Learning Outcome 2: Drawing on skills developed in class, students will be able to critically evaluate, analyze, and generate arguments, bodies of evidence, and/or claims, including their own.

Students apply the critical thinking techniques of this course to a large variety of algorithms, covering all the major categories. They write many different algorithms and evaluate them for correctness and time complexity.

Program Learning Outcomes

Study Abroad Sydney Program Outcome:

The student will “demonstrate knowledge of Australian culture and society with respect to a combination of the following areas: Australian politics, industry, science and technology, economics, social policy, environmental policy, literature and the arts, film, marketing, advertising, and mass media”.

Study Abroad Sydney Program Outcome

The student will “demonstrate knowledge of Australian culture and society with respect to a combination of the following areas: Australian politics, industry, science and technology, economics, social policy, environmental policy, literature and the arts, film, marketing, advertising, and mass media”.

Programming Environment

We will use Python and Jupyter Notebooks to demonstrate and test the algorithms. Students can utilize Google Colab for their work.

Books and Other Course Materials

Algorithm Design, by Kleinberg and Tardos. ISBN 0-321-29535-8.

Useful additional resources:

- Cormen, Leiserson, Rivest, and Stein. *Introduction to Algorithms*, 3rd ed. MIT Press.

- J. Erickson. *Algorithms,* 2019. Available from http://algorithms.wtf/ See also the extensive exercises on the website. ● Mathematics for Computer Science by Eric Lehman, Tom Leighton, and Albert Meyer. (Useful background on discrete mathematics.)

Assignments and Grading

Final Grades

	All Grades out of 100 Points		
Grade	**Max**	**Avg**	**Min**
F	59.4	50	0.0
D	69.4	65	59.5
C-	72.4	72	69.5
C	76.4	75	72.5
C+	79.4	78	76.5
B-	82.4	82	79.5
B	86.4	85	82.5
B+	89.4	88	86.5
A-	93.4	92	89.5
A	100	96	93.5

Assessment and Marking Weighting and due date

In this course, students will participate in an online quiz, complete three assignments, take a final examination, and go on a field visit.

- **Class Participation based on topics – 5%**
- **Online Quiz – 15%**
- **Final Examination – 20%**

All submissions are due by Wednesday at 8 PM with the BU Sydney Assignnment coversheet during the designated submission week. Students are welcome to submit their assignments earlier if needed.

Assessment	Weighting	Submission week	Details
Assessment 1	20%	Week 3	**This is an in-class presentation.** Prepare a presentation investigating data structures and abstract data types. Analyze their theoretical foundations, design principles, and implementation considerations. Evaluate their applicability in solving complex computational problems. Provide a detailed analysis of their performance, trade-offs, and potential optimizations. The presentations should demonstrate advanced knowledge and critical thinking skills in the field of data structures and abstract data types. The presentation must include slides, with the following slide headings: (1) Introduction (2) Data Structures Overview (3) Importance of Data Structures (4) Data Structures Operations (5) Investigating Different Data Structures (6) Conclusion Students must present their presentation in class in week 3. 2. The presentation should exhibit depth and sophistication in discussing data structures and abstract data types. 3. It should demonstrate a comprehensive understanding of the theoretical foundations and practical aspects. 4. Analyze the design principles and implementation strategies for different data structures and abstract data types. 5. Evaluate their strengths, weaknesses, and trade-offs in solving complex computational problems. 6. Provide insights into optimizing data structures and abstract data types for specific scenarios. 7. Use scholarly references to support the analysis and provide a well-rounded perspective. 8. Present the ideas logically and coherently, adhering to academic writing conventions. 9. Please refer to the assessment marking guide to address all the assessment criteria.

Assessment 2	20%	Week 5	In this assignment, students will submit their **Jupyter notebook**. Students will implement a basic neural network in Python using **TensorFlow** and explore how artificial intelligence (AI) leverages data structures and algorithms. The task involves setting up the environment, loading a simple dataset like MNIST or Iris, and defining a feedforward neural network with one or two hidden layers using the Sigmoid activation function. Students will train the network by splitting the dataset, compiling the model with a loss function and optimizer, and evaluating its performance, visualizing results with plots of training and validation accuracy and loss. The Jupyter notebook deliverable will include both the code implementation and a discussion section, explaining neural network concepts, the role of data structures like arrays and matrices in representing data and weights, and the importance of algorithms such as backpropagation and gradient descent in training neural networks. Additionally, the discussion should highlight practical AI applications in fields like image recognition and natural language processing. Evaluation will focus on the correctness of the implementation, the depth and clarity of the theoretical discussion, and the organization and presentation of the notebook.
Assessment 3	20%	Week 7	**This assignment is a report** that explores the impact of quantum computing on cryptography, focusing on its potential to revolutionize and challenge traditional cryptographic methods. Quantum computing leverages principles of superposition and entanglement, allowing it to perform computations exponentially faster than classical computers. This poses a significant threat to widely used cryptographic systems, such as RSA and ECC, which rely on the difficulty of factoring large numbers and solving discrete logarithm problems—tasks that quantum algorithms like Shor's algorithm can solve efficiently. The report will also discuss how quantum computing necessitates the development of quantum-resistant cryptographic methods, known as post-quantum cryptography, to secure data against future quantum attacks. Additionally, the report will highlight the promising role of quantum key distribution (QKD) in enhancing security through the principles of quantum mechanics, ensuring that any eavesdropping attempts are detectable. By examining these aspects, the report aims to provide a comprehensive understanding of quantum computing's dual role in both threatening and advancing cryptographic practices.

Community of Learning: Class and University Policies

Course atmosphere, diversity and inclusion:

We intend to provide a positive and inclusive atmosphere in class and on the associated virtual platforms. Students from a wide range of backgrounds and with a diverse set of perspectives are welcome. We ask that students treat each other with thoughtfulness and respect, and do their part to make all their peers feel welcome. Your suggestions are encouraged and appreciated. Please let us know ways to improve the effectiveness of the course for you personally or for other students or student groups.

Course Matters

Attendance at all classes is mandatory.

Any absence for medical reasons or other misadventure must be supported by a medical certificate or a letter offering a satisfactory explanation. Strict penalties apply, on a pro rata basis, for any unapproved absence. Missing one class without reason would attract as much as a 10% penalty.

Statement on Plagiarism

All students are responsible for having read the Boston University statement on plagiarism, which is available in the Academic Conduct Code. Students are advised that the penalty against students on a Boston University program for cheating on examinations or for plagiarism may be "… expulsion from the program or the University or such other penalty as may be recommended by the Committee on Student Academic Conduct, subject to approval by the Dean".

Academic Conduct

Academic standards and the code of academic conduct are taken very seriously by the University, the College of Arts and Sciences, and the Department of Computer Science. Course participants must adhere to the CAS Academic Conduct Code. Please take the time to review this document if you are unfamiliar with its contents.

Any case of academic misconduct, including but not limited to plagiarism and submission of work that was not solved by you – be that a tutor, friend, web resource, AI, etc., will be reported to the College and will also carry a grading penalty.

Late Work

In general, there will be no extensions granted for any coursework. The exception is where there are clear and acceptable reasons for late submission. In this case a written statement outlining any serious illness or misadventure together with supporting documentation (e.g. medical certificates) must be provided or a strict penalty of 5% per day will apply.

Tentative schedule

This schedule is for informational purposes only and is subject, and is likely, to change as we progress through the semester.

Week	**Topic**
1	Fundamental Data Structures (Stack, Queues, Arrays, Linked Lists)
2	Advanced-Data Structures (Heap, Trees, Graphs)
3	In Class Presentation (Assignment 1)
4	Machine Learning and Neural Networks
4	Field Trip on 13th Friday September to **University of Technology Sydney (UTS)**.
5	Algorithm Analysis and Complexity
6	Algorithm Design Techniques
7	Applications of Data Structures and Algorithms

Appendix

https://www.geeksforgeeks.org/learn-data-structures-and-algorithms-dsa-tutorial/

https://github.com/TheAlgorithms/Python