MH1403 Algorithms and Computing

Academic Year	2018/2019	**Semester**	2
Course Coordinator	Guo Jian, guojian@ntu.edu.sg		
Course Code	MH1403		
Course Title	Algorithms and Computing		
Pre-requisites	PS0001 Introduction to Computational Thinking		
No of AUs	3		
Contact Hours	Lecture: 13 hrs Mini-Lecture/Tutorial/Labs: 39 hrs		
Proposal Date	24/05/2018		

Course Aims

This course aims to give you a systematic introduction to data structures and algorithms for constructing efficient computer programs. Emphasis is on data abstraction issues in the program development process, and on the design of efficient algorithms. Simple algorithmic paradigms such as greedy algorithms, divide-and-conquer algorithms and dynamic programming will be introduced. Elementary analyses of algorithmic complexities will also be taught.

Intended Learning Outcomes (ILO)

By the end of the course, you should be able to:

1. Implement data structures in Python.
2. Analyze the complexity of algorithms.
3. Design and implement efficient algorithms for given applications.
4. Solve large problems systematically and effectively.
5. Work as a team and collaborate to manage larger programming projects

Course Content

- Run Time Notations and Complexities
- Algorithm Analysis
- Data Structures: List, Stack, Queue, Tree, Binary Search Tree
- Sorting: Quick Sort, Merge Sort
- Greedy Methods
- Divide-and-Conquer
- Dynamic Programming

Assessment (includes both continuous and summative assessment)

Component	Course ILO Tested	Related Programme LO or Graduate Attributes	Weighting	Team/Individual	Assessment rubrics
1. Individual Project	1,2,3	A1, A2, A3, A4, B1, B2, B3, B4, C2, D1	10%	Individual	Appendix 1
2. Group Project	1,2,4,5	A1, A2, A3, A4, B1, B2, B3, B4, C2, D1, E1	10%	Team	Appendix 2
3. Midterm Exam	1,2,3,4	A1, A2, A3, A4, B1, B2, B3, B4, D1	20%	Individual	Appendix 3
4. Final Exam	1,2,3,4	A1, A2, A3, A4, B1, B2, B3, B4, D1	60%	Individual	Appendix 4
Total			100%		

Formative feedback

You will be receiving formative feedback verbally during tutorials and through written grading of your lab assignments. You will receive summative group feedback on the exam following the conclusion of the course.

Learning and Teaching approach

Approach	**How does this approach support you in achieving the learning outcomes?**
Lecture, Tutorial and Lab Practice	Lectures are to introduce the basic knowledge on data structure, algorithms and complexity analysis. The concept of each lecture will be enhanced by the tutorial and lab sessions after the lectures.
Tutorial	Tutorials are divided into two parts: introduction to the programming of the data structures, and solution solving. You are expected to solve the tutorial questions by yourselves before coming to the tutorial sessions, and solutions will only be provided after that.
Lab Practice	Lab sessions will be on the format of both individual and group projects, you are expected to solve the given project by Python programming languages. Lab tutors will be present at the lab sessions only to guide you in finding solutions by yourselves

Reading and References

- Textbook: Michael T. Goodrich and Roberto Tamassia, Algorithm Design and Applications, WILEY, 2014 (978-1-118-33591-8)
- Reference: Huang Guangbin and Ng Jim Mee, Data structures and algorithms, Pearson Education South Asia, 2007 (9789810679149)

Both books will be available at NTU libraries.

Course Policies and Student Responsibilities

You are expected to attend the lectures and the tutorial sessions, and to take all scheduled assignments and projects by due dates. Not submitting a lab assignment or project before the corresponding deadline will be counted as no submission. You are expected to take responsibility to follow up with course notes, assignments and course related announcements they have missed.

Academic Integrity

Good academic work depends on honesty and ethical behaviour. The quality of your work as a student relies on adhering to the principles of academic integrity and to the NTU Honour Code, a set of values shared by the whole university community. Truth, Trust and Justice are at the core of NTU's shared values.

As a student, it is important that you recognize your responsibilities in understanding and applying the principles of academic integrity in all the work you do at NTU. Not knowing what is involved in maintaining academic integrity does not excuse academic dishonesty. You need to actively equip yourself with strategies to avoid all forms of academic dishonesty, including plagiarism, academic fraud, collusion and cheating. If you are uncertain of the definitions of any of these terms, you should go to the academic integrity website for more information. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Collaboration is encouraged for your homework because peer-to-peer learning helps you understand the subject better and working in a team trains you to better communicate with others. As part of academic integrity, crediting others for their contribution to your work promotes ethical practice.

You must write up your solutions by yourself and understand anything that you hand in. You are not allowed to copy/paste code directly from your collaborator. If you use outside source, you must reference it in your solution. If you are found to have copied directly from your collaborator or from an external source without proper reference, you may face grade penalty, course failure, official academic misconduct report or even exclusion for the most serious cases (see http://www.ntu.edu.sg/ai/Pages/academic-integrity-policy.aspx).

Course Instructors

Instructor	Office Location	Phone	Email
Guo Jian	SPMS-MAS-05-01	6514 8399	guojian@ntu.edu.sg

Planned Weekly Schedule

Week	Topic	Course ILO	Readings/ Activities
1	Introduction to the course	1,2,3,4	Lecture notes
2	Run Time and Complexities	1,2,3,4	Lecture notes
3	Algorithm Analysis	1,2,3,4	Lecture notes
4	Stack	1,2,3,4	Lecture notes
5	Queue	1,2,3,4	Lecture notes
6	List	1,2,3,4	Lecture notes
7	Tree	1,2,3,4	Lecture notes
8	Binary Search Tree	1,2,3,4	Lecture notes
9	Merge Sort	1,2,3,4,5	Lecture notes
10	Quick Sort	1,2,3,4,5	Lecture notes
11	Greedy Method	1,2,3,4,5	Lecture notes
12	Divide-and-Conquer	1,2,3,4,5	Lecture notes
13	Revision	1,2,3,4	Lecture notes

Appendix 1: Assessment Criteria for Individual Project

<u>Assessment Component CA 1: Individual Project (10%)</u>

Learning Outcomes Related to this Project:

LO1. Implement data structures in Python

LO2. Analyze the complexity of algorithms

LO3. Design and implement efficient algorithms for given applications

Grading Criteria	Exceptional	Effective	Acceptable	Developing
Quality of the project	Very efficient and clean implementation of the project (LO1); excellent algorithm design and good intuition on solving the various tasks (LO2,LO3)	Efficient and clean implementation of the project (LO1); proper algorithm design and appropriate intuition on solving the various tasks (LO2,LO3)	Somewhat efficient and clean implementation of the project (LO1); acceptable algorithm design and moderate intuition on solving the various tasks (LO2,LO3)	Inefficient or messy implementation of the project (LO1); wrong algorithm design and limited intuition on solving the various tasks (LO2,LO3)
Clarity of the code produced	Very clear code produced with extensive comments (LO1).	Mostly clear code produced and good amount of comments (LO1).	Somewhat clear code produced and adequate amount of comments (LO1).	Unclear code produced and too few comments (LO1)

Appendix 2: Assessment Criteria for Group Project

Assessment Component CA 2: Group Project (10%)

Learning Outcomes Related to Group Project:

LO1. Implement data structures in Python

LO2. Analyze the complexity of algorithms

LO4. Solve large problems systematically and effectively.

LO5. Work as a team and collaborate to manage larger programming projects

Learning outcome of the programme:

C2. Work in teams on complicated projects that require applications of Mathematics, and communicate the results verbally and in written form

Grading Criteria	Exceptional	Effective	Acceptable	Developing
Validity of the implementation	Fully valid implementations of the problems to solve (LO1) , the code compiles and runs as expected	Mostly valid implementations of the problems to solve (LO1)	Somewhat valid implementations of the problems to solve (LO1)	Invalid implementations of the problems to solve (LO1)
Analysis of data	Excellent extraction of the data from the given problem (LO1)	Good extraction of the data from the given problem (LO1)	Moderate extraction of the data from the given problem (LO1)	Mis-extraction of the data from the given problem (LO1)
Algorithm Efficiency	Excellent algorithm design and good intuition on solving the various tasks (LO2, LO4)	Proper algorithm design and appropriate intuition on solving the various tasks (LO2, LO4)	Acceptable algorithm design and moderate intuition on solving the various tasks (LO2, LO4)	Wrong algorithm design and limited intuition on solving the various tasks (LO2, LO4)
Clarity of the code produced	Very clear code produced with extensive comments (LO4 and C2).	Mostly clear code produced and good amount of comments (LO1, and C2).	Somewhat clear code produced and adequate amount of comments (LO1, and C2).	Unclear code produced and too few comments (LO1, and C2).
Collaboration	Very consistent and balanced separation of the various tasks of the project among the students (LO5 and C2)	Consistent and balanced separation of the various tasks of the project among the students (LO5 and C2)	Somewhat consistent and balanced separation of the various tasks of the project among the students (LO5 and C2)	Inconsistent or unbalanced separation of the various tasks of the project among the students (LO5 and C2)

Appendix 3: Assessment Criteria for Mid-Term Exam

Assessment Component ME: Mid-Term Exam (20%)

Learning Outcomes Related:

LO1. Implement data structures in Python
LO2. Analyze the complexity of algorithms
LO3. Design and implement efficient algorithms for given applications
LO4. Solve large problems systematically and effectively.

Grading Criteria	**Exceptional**	**Effective**	**Acceptable**	**Developing**
Validity of the implementation	Fully valid implementations of the problems to solve (LO1) , the code compiles and runs as expected	Mostly valid implementations of the problems to solve (LO1)	Somewhat valid implementations of the problems to solve (LO1)	Invalid implementations of the problems to solve (LO1)
Analysis of data	Excellent extraction of the data from the given problem (LO2)	Good extraction of the data from the given problem (LO2)	Moderate extraction of the data from the given problem (LO2)	Mis-extraction of the data from the given problem (LO2)
Algorithm Efficiency	Excellent algorithm design and good intuition on solving the various tasks (LO3,LO4)	Proper algorithm design and appropriate intuition on solving the various tasks (LO3,LO4)	Acceptable algorithm design and moderate intuition on solving the various tasks (LO3,LO4)	Wrong algorithm design and limited intuition on solving the various tasks (LO3,LO4)

Appendix 4: Assessment Criteria for Final Exam

Assessment Component FE: Final Exam (60%)

Learning Outcomes Related:

LO1. Understand and implement data structures

LO2. Analyze the complexity of algorithms

LO3. Design and implement efficient algorithms for given applications

LO4. Solve large problems systematically and effectively.

Grading Criteria	Exceptional	Effective	Acceptable	Developing
Validity of the implementation	Fully valid implementations of the problems to solve (LO1), the code compiles and runs as expected	Mostly valid implementations of the problems to solve (LO1)	Somewhat valid implementations of the problems to solve (LO1)	Invalid implementations of the problems to solve (LO1)
Analysis of data	Excellent extraction of the data from the given problem (LO2)	Good extraction of the data from the given problem (LO2)	Moderate extraction of the data from the given problem (LO2)	Mis-extraction of the data from the given problem (LO2)
Algorithm Efficiency	Excellent algorithm design and good intuition on solving the various tasks (LO3,LO4)	Proper algorithm design and appropriate intuition on solving the various tasks (LO3,LO4)	Acceptable algorithm design and moderate intuition on solving the various tasks (LO3,LO4)	Wrong algorithm design and limited intuition on solving the various tasks (LO3,LO4)