Course Syllabus

ICCS 321 — Data Structures and Algorithm Analysis (4-0 credits)

1 Basic Course Information

Instructor: Dr. Kanat Tangwongsan (`kanat.tan@mahidol.ac.th`)

Prerequisite(s): ICCS 201 or instructor's permission

Textbook: We will be very loosely following

> Bradley N. Miller, David L. Ranum: *Problem Solving with Algorithms and Data Structures Using Python* (2nd ed.). Franklin, Beedle & Associates 2011.

which is available online free of charge for academic use.

2 Course Description

This course aims to teach methods for designing, analyzing, and programming algorithms and data structures. It focuses on fundamental concepts that can be applied across problem domains, programming languages, and computer architectures. There will be a significant programming component, which will be primarily in Python. ***We will be introducing Python concepts along with other material.***

Topics covered in this class include: problems and abstractions, contracts and interfaces, performance analysis (asymptotic notation and analysis), basic data structures (stacks, queues, lists, double-ended queues), sorting, searching (linear search, binary search, collection), divide and conquer techniques, unit testing, (search) trees, hashing, regular expressions, and basic graph representations.

3 Course Objectives and Goals

At the completion of this course, students will be able to select and apply appropriate data structures and algorithms for different applications. In particular, the students will

(1) understand the concepts of linear data structures and collections, and methods for ensuring correctness of algorithms/programs;

(2) understand how to design, analyze, and implement basic algorithms; and

(3) be able to efficiently manipulate data using data structuring and algorithmic techniques.

4 Course Outline

Week 1: Course overview; Python basics; specifying, verifying, and reasoning about contracts

Week 2: Searches (linear search, binary search), and reasoning about performance and correctness

Week 3: Sorting, reasoning about recursion, induction

Week 4: Divide and conquer algorithms, more sorting

Week 5: * **Midterm I** *, abstractions and interfaces, and their relationships to performance

Week 6: Stacks, queues, lists, double-ended queues; unit testing

Week 7: Big-Oh and simple asymptotic analysis; priority queues

Week 8: Binary search trees, balanced binary search trees, and other search trees

Week 9: * **Midterm II** *, hashing, and hash tables

Week 10: Regular expressions, graph representations, basic graph algorithms

Week 11: Bonus topics

5 Evaluation

5.1 Grading Policy

Assignments (5–8 sets)	25%
2 Midterms	40%
Final	30%
Participation & Quizzes	5%

5.2 Letter Grades

There is no exact formula for assigning letter grades, but as a general guideline, we will use the following cutoffs:

90+	A	70 – 74	C
85 – 89	B+	65 – 69	D+
80 – 84	B	60 – 64	D
75 – 79	C+	0 – 60	F

5.3 Late Assignments

Assignments are due electronically at 11:59PM Bangkok time unless otherwise stated on the assignment. You are encouraged to hand them in well ahead of the deadline.

You are allotted **FIVE (5) late days for the term** at no grade penalty. At most **TWO (2) late days may be used per assignment**. If you have used up these five days or used more than two for a given assignment, your score will be reduced by 25% off of the total (not your score) per late day used. Except in extraordinary circumstances (e.g., a medical emergency), no late homework will be accepted beyond the late date.

5.4 Academic Integrity

Students are expected to maintain the highest standards of academic integrity. Academic dishonesty will *not* be tolerated and will be reported to the Academic Dishonesty committee.

6 References

Michael T. Goodrich, Roberto Tamassia: *Data structures and algorithms in Java* (3rd ed.). Wiley 2003.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: *Introduction to Algorithms* (3rd ed.). The MIT Press 2009.