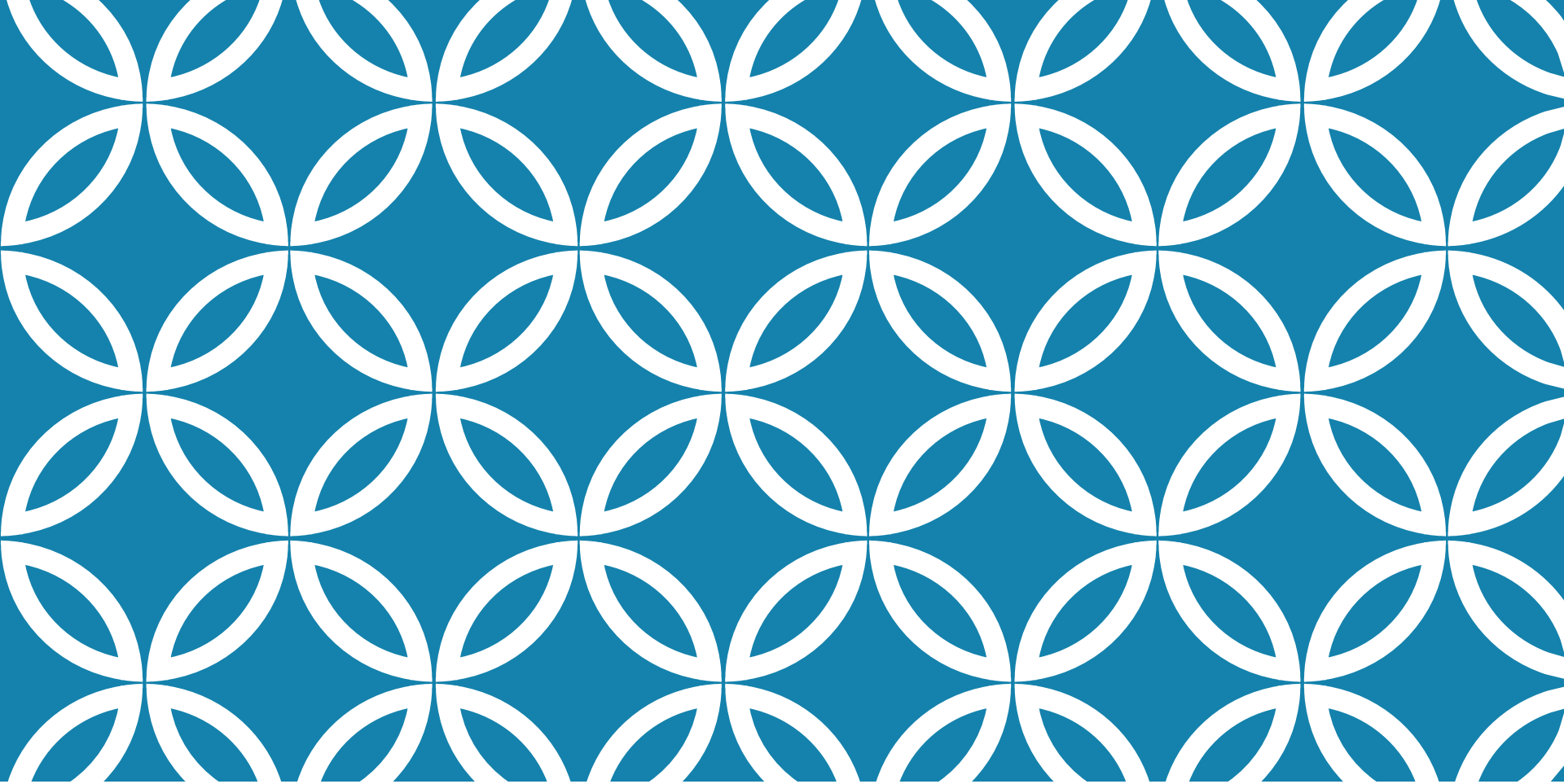

# AST20105 DATA STRUCTURES & ALGORITHMS

## COURSE LOGISTICS

Garret Lai

# LEARNING OBJECTIVES

Upon successful completion of this course, students will be able to:

1. Identify the basic abstract data types and algorithms for storing, retrieval, and searching;
2. Design and implement linked list, stack, queue, tree and graph structures;
3. Analyse algorithms, including different sorting methods for appropriate usage and fine tuning;
4. Perform execution measurement and performance evaluation with asymptotic notation;
5. Apply and select the appropriate abstract data types for different applications.

# AREA COVERED

## Types of data structure

- Array and Link list
- Tree and Graph
- Hash, and more…

## Algorithm Efficiency Analysis

- Big-O, Big-Omega

## Sorting Algorithms

- Merge sort, insertion sort, quick sort, bubble sort, etc

## Design of Algorithm

- Time and space limitation

# WEEKLY SCHEDULE (TENTATIVE)

| WK | Lecture | Lab |
|---|---|---|
| 1 | • Course Introduction<br>• Introduction to data structure<br>• Importance of algorithm analysis | • Quick revision of C++ |
| 2 | • Revision on mathematics related computing complexity<br>• Algorithm complexity definition | • C++ template study |
| 3 | • Algorithm complexity analysis | • Recursive function |
| 4 | • Programming design approaches and their complexity | • Analysis of algorithm |
| 5 | • Array and linked list<br>• Complexity analysis on array and linked list structures | • Array and link list<br>• Release of project |
| 6 | • Stack and queue | • Stack and queue |

# WEEKLY SCHEDULE (TENTATIVE)

| WK | Lecture | Lab |
|---|---|---|
| 7 | • Tree, binary search tree and their operations<br>• AVL Tree and their operations | • Binary Tree |
| 8 | • AVL Tree, B-tree and their operations | • AVL Tree |
| 9 | • Graphs I | • B-tree<br>• Mid-term Test (Practical) on Sat |
| 10 | • Graphs II | • Analysis of algorithm |
| 11 | • Sorting algorithms and their complexity analysis (I)<br>• Selection sort, Insertion sort, Merge sort, Partitioning and quick sort | • Sorting algorithm I |

# WEEKLY SCHEDULE (TENTATIVE)

| WK | Lecture | Lab |
|---|---|---|
| 12 | • Comparison-based sorting continued<br>• Non-comparison-based sorting: Counting sort, bucket sort and radix sort | • Project prototype demo |
| 13 | • Non-comparison-based sorting: Counting sort, bucket sort and radix sort<br>• Revision | • Non-comparison-based sorting<br>• Project due |

# ASSESSMENT SCHEDULE

## Examination (60%)

- 2 hours closed book

## Lab Exercises (15%)

- Practice of theories learnt in classes

## Test (10%)

- Practical test

## Programming Assignments (15%)

- Working in groups

# SHOULD YOU NEED HELP…

Consultation hours (will be stated next week)

By appointment (if consultation hours don't fit you)

Tel: 34425780

Room: AC2 6433

Email me: yikhglai2@cityu.edu.hk

# Q & A