

CSCI 204: Data Structures & Algorithms

*Revised by Xiannong Meng based on
textbook author's notes*

Course Summary

What We Started With – Four Areas

- Data Structures
 - Particular way to organize and manipulate data, e.g., lists, arrays, stacks, queues, and trees
- Data Abstraction
 - Data type *properties* (independent of implementation)
- Algorithms & Algorithm Analysis
 - Common algorithms, searching, sorting
- Advanced Python Programming
 - OOP, classes, inheritance, exceptions, etc.

Data Structures and Data Abstraction (1)

- Dictionary
- Python list
- Linked list
 - Singly linked list
 - Doubly linked list
- Stack ADT
 - Linked list based stack
 - Array based stack

Data Structures and Data Abstraction (2)

- Queue ADT
 - Linked list based queue
 - Array based queue
 - Priority queue
- Binary tree ADT
 - Binary search tree
 - Heap
 - General binary tree
- Hash tables
 - Closed hashing (what we discussed), conflict resolution

Algorithms and Analyses (1)

- Linear search $O(n)$
- Binary search $O(\log n)$
- Bubble sort $O(n^2)$
- Selection sort $O(n^2)$
- Insertion sort $O(n^2)$
- Quicksort $O(n \log n)$ average case
- Merge sort $O(n \log n)$
- Heap sort $O(n \log n)$ average case

Algorithms and Analyses (2)

- We use Big-Oh notation to classify the complexity of the algorithms – time needed to complete the run of the algorithm for a given input size of n
- $O(1)$, $O(\log n)$, $O(n \log n)$, $O(n^2)$, ... $O(2^n)$
- The time needed to complete the algorithm grows to the right. An algorithm with a complexity of $O(2^n)$ is impractical to solve any real problems

Advanced Python and Programming

- Class
- Inheritance
- Recursion
- Exception
- User defined modules
 - Import from user defined modules
- Python functions as parameters