

New York University Tandon School of Engineering
EL-GY 9343: Special Topic Course in Telecom Networks
Data Structure and Algorithm Session B & INET
Fall 2020

Instructor: Yong Liu

Course Prerequisites

- 1) Basic knowledge of fundamental data structures.
 - 2) Basic programming language skills, such as C/C++, Java, Python
- If you are not sure you have the proper preparation, you must talk to me before taking this course. Additionally, you should not take this course if you have taken a similar course, such as CS6033 with a 'B' or better grade.

Course Description

- Review of basic data structures and mathematical tools.
- Data structures: priority queues, binary search trees, balanced search trees.
- Algorithm design and analysis techniques illustrated in searching and sorting: heapsort, quicksort, sorting in linear time, medians and order statistics.
- Design and analysis techniques: divide and conquer, dynamic programming, greedy algorithms.
- Graph algorithms: elementary graph algorithms (breadth-first search, depth-first search, topological sort, connected components, strongly connected components), minimum spanning trees, shortest paths.
- Brief introduction of complexity and NP-completeness.

Textbook

Cormen, Leiserson, Rivest, and Stein,
Introduction to Algorithms, 3rd Edition, MIT Press, 2009;
ISBN-13: 9780262033848; The paperback international version has ISBN-13
9780262533058. It is known as CLRS.

In-class Session Meeting: Monday 11am – 1:30pm, Pfizer Auditorium, 5 MTC

Virtual Office Hours: Monday 4pm to 5pm

Course Work and Grading: Your final grade will be determined roughly as follows:

Homework	10%
Midterm	40%
Final	50%

Tentative Schedule

- Week1: (09/09) Introduction to algorithm: correctness and performance. Best-, worst-, and average-case performance. Asymptotic notation: big-O, big- Ω , and big- Θ ; little-o, and little- ω .
- Week 2: (09/14) Recurrence and solving methods: iteration, substitution and master theorem
- Week 3 (09/21) Divide and conquer algorithms, introduction to sorting: insertion sort, bubble sort
- Week 4 (09/28) Sorting: MergeSort, Heap and HeapSort,
- Week 5: (10/05) Sorting: quick sort, randomized algorithms, lower bound for comparison sorting, counting sort and radix sort, order statistics and selection
- Week 6: (10/12), Hashing and Universal Hashing, Binary search trees
- Week 7: (10/19) Binary search trees (cont.d), midterm review
- Week 8: (10/26, Tentative) Midterm
- Week 9: (11/02) Graph basics, Breath-First Search, Depth-First Search
- Week 10: (11/09) Directed-acyclic graph and topological ordering, strongly connected components,
- Week 11: (11/16) Intro to dynamic programming, greedy algorithm
- Week 12: (11/23) Greedy algorithm, Huffman coding, Minimum Spanning Tree
- Week 13: (11/30) Single-source shortest paths, all-pairs shortest paths
- Week 14: (12/07) NP-Completeness and Final Review
- Week 15: (12/14 – 12/21, TBD) Final