Lecture #1: Tuesday, 2 April 2002
Topics: Course Outline

Course Outline

The chapter and section references below are for the *first edition* of CLR.

April

Tue	2	Administrivia. Introduction: models of computation, O-notation. *Reading:* Section 1.1, Chapter 2.
Thu	4	Insertion sort and mergesort, divide and conquer, recurrences. *Reading:* Sections 1.2-3. Section 4.1
Tue	9	Quicksort, Strassens' algorithm, more on summations and recurrences. *Reading:* Sections 8.1-2, 31.2. Sections 4.2-3, (Chapter 3 should be read as needed during the quarter.)
Thu	11	Randomized algorithms: randomized quicksort, probability. *Reading:* Sections 6.1-3, 8.3-4, (Chapter 5 should be read as needed during the quarter.)
Tue	16	Sorting: median, order statistics. *Reading:* Chapter 10.
Thu	18	Sorting: heapsort, priority queues, set manipulation. *Reading:* Chapter 7.
Tue	23	Sorting: lower bounds, counting sort, radix sort. *Reading:* Chapter 9.
Thu	25	Data structures: hashing, collision resolution, chaining, universal hashing, open addressing. *Reading:* Chapter 12.
Tue	30	Data structures: binary search trees, tree walks, relation to quicksort. *Reading:* Chapter 13.

May

Thu	2	Data structures: red-black trees, rotations, insertion, deletion. *Reading:* Chapter 14.
Tue	7	Mid-term examination, in class, closed book.
Thu	9	Augmenting data structures: dynamic order statistics, interval trees. Programming Problem handed out. *Reading:* Chapter 15.
Tue	14	Dynamic programming: optimal binary search trees, longest common subsequence. *Reading:* Chapter 16.
Thu	16	Greedy algorithms: activity selection. Introduction to graph algorithms: representation, breadth-first search. *Reading:* Section 17.1-3, 23.1-2.
Tue	21	Graph algorithms: minimum-spanning tree algorithms, Prim's algorithm, Kruskal's algorithm. *Reading:* Chapter 24.
Thu	23	Graph algorithms: depth-first search, topological sort. *Reading:* Section 23.3-4.
Tue	28	Graph algorithms: Single-source shortest paths, Dijkstra's algorithm, Bellman-Ford algorithm, difference constraints. *Reading:* Chapter 25.
Thu	30	Graph algorithms: all-pairs shortest paths, matrix multiplication, Floyd-Warshall algorithm. *Reading:* Chapter 26.

June

Tue	4	Flow networks; the Ford-Fulkerson Algorithm; Bipartite matching. *Reading:* Chapter 27. Programming Problem due. *All homeworks due by this date.*
Thu	6	Special end-of-class lecture. Course evaluation. *Reading:* None.

The chapter and section references below are for the *second edition* of CLRS.

April

Tue	2	Administrivia. Introduction: models of computation, O-notation. *Reading:* Section 1.1, Chapter 3.
Thu	4	Insertion sort and mergesort, divide and conquer, recurrences. *Reading:* Sections 1.2. Chapter 2. Section 4.1
Tue	9	Quicksort, Strassens' algorithm, more on summations and recurrences. *Reading:* Sections 7.1-2, 31.2. Sections 4.2-3, (Chapter 3 should be read as needed during the quarter.)
Thu	11	Randomized algorithms: randomized quicksort, probability. *Reading:* Chapter 5, 7.3-4, (Chapter 5 should be read as needed during the quarter.)
Tue	16	Sorting: median, order statistics. *Reading:* Chapter 9.
Thu	18	Sorting: heapsort, priority queues, set manipulation. *Reading:* Chapter 6.
Tue	23	Sorting: lower bounds, counting sort, radix sort. *Reading:* Chapter 8.
Thu	25	Data structures: hashing, collision resolution, chaining, universal hashing, open addressing. *Reading:* Chapter 11.
Tue	30	Data structures: binary search trees, tree walks, relation to quicksort. *Reading:* Chapter 12.

May

Thu	2	Data structures: red-black trees, rotations, insertion, deletion. *Reading:* Chapter 13.
Tue	7	Mid-term examination, in class, closed book.
Thu	9	Augmenting data structures: dynamic order statistics, interval trees. Programming Problem handed out. *Reading:* Chapter 14.

Tue	14	Dynamic programming: optimal binary search trees, longest common subsequence. *Reading:* Chapter 15.
Thu	16	Greedy algorithms: activity selection. Introduction to graph algorithms: representation, breadth-first search. *Reading:* Section 16.1-3, 22.1-2.
Tue	21	Graph algorithms: minimum-spanning tree algorithms, Prim's algorithm, Kruskal's algorithm. *Reading:* Chapter 23.
Thu	23	Graph algorithms: depth-first search, topological sort. *Reading:* Section 22.3-4.
Tue	28	Graph algorithms: Single-source shortest paths, Dijkstra's algorithm, Bellman-Ford algorithm, difference constraints. *Reading:* Chapter 24.
Thu	30	Graph algorithms: all-pairs shortest paths, matrix multiplication, Floyd-Warshall algorithm. *Reading:* Chapter 25.

June

Tue	4	Flow networks; the Ford-Fulkerson Algorithm; Bipartite matching. *Reading:* Chapter 26. Programming Problem due. *All homeworks due by this date.*
Thu	6	Special end-of-class lecture. Course evaluation. *Reading:* None.