Data Structures course info lectures exams

R. Inkulu at cse.iitg in Fall 2019

Asymptotic notation [CLRS]: 43-52

Introduction [CLRS]: 1-14, 23-29, 147-148; [HSA]: 22-40

Elementary data structures

- Array [HSA]: 18-21, 51-55, 67, 74; [CLRS]: 230-231
- Linked list [CLRS]: 236-240; [HSA]: 145-149, 186-189
- Appl: polynomials and sparse matrices [HSA]: 64-83, 160-168, 178-185 --- AR
- Stack [CLRS]: 232-234; [HSA]: 107-113
- Appl: evaluating expressions [HSA]: 127-136
- Queue [CLRS]: 234-235; [HSA]: 114-120
- Trees and their traversals [CLRS]: 246-248; [HSA]: 205-211, 216-222

Searching in a sorted array

- Linear search, binary search [HSA]: 12-13

More analysis techniques

- Probabilistic analysis: hiring [CLRS]: 114-116, 120-121, 139-141, 1154-1156
- Expected analysis: randomized hiring [CLRS]: 122-124, 126-128
- Amortized analysis: stack with multipop, binary counter [CLRS]: 451-461
- Amortized analysis: dynamic array that only expands [CLRS]: 463-466

Comparison sort

- Bubble sort, selection sort [wiki]: 1, 2
- Insertion sort [CLRS]: 16-22; [wiki]
- Shellsort (with Shell's sequence) [W]: 296-298; wiki
- Probabilistic analysis: sorting and searching [R]: 480-481, 482-484
- Mergesort (divide and conquer) [CLRS]: 29-37
- Quicksort (divide and conquer) [CLRS]: 170-176, 180-181
- Randomized quicksort [CLRS]: 177-179, 181-184
- Also see heapsort noted below
- Bucket sort [CLRS]: 200-204

Sorting with no comparisons

- Counting sort [CLRS]: 194-196
- Radix sort [CLRS]: 197-199

Selection

- Minimum and maximum [CLRS]: 213-215
- Hoare's Las Vegas algo [CLRS]: 215-219
- Blum et al.'s worst-case linear time algo [CLRS]: 220-222
- A Monte Carlo algorithm [MU]: 57-62 --- not covered this time

Worst-case lower bound

- Decision trees: comparison sort [CLRS]: 191-193
- Adversary arguments: min, min and max, median [note]

Dictionary

- Algo for basic operations [CLRS]: 286-299; [HSA]: 236-237
- Expected height of a randomly built BST [CLRS]: 299-303; [Jensen's ineq]
- AVL tree [W]: 144-153; [insertcases: 1, 2]; [deletealgo]
- 2-3-4 tree and its generalization [CLRS]: 488-502; [HSA]: 551-552
- Red-Black tree [CLRS]: 308-329; [HSA]: 514-515; [234vsRB]
- Splay tree [HSA]: 518-524

Dictionary for strings

- Tries sketched [HSA]: 561-564, 571-574, 577-580, 584-592
- Suffix trie, suffix array, LCP array [HSA]: 593-600; [note]
- Ukkonen's algo for suffix tree construction [Gus]: 94-107 --- not covered this time

Randomized dictionary: Hashing

- Intro to open hashing [CLRS]: 253-260
- Universal hashing [CLRS]: 265-268; [note]
- Perfect hashing and FKS algo [CLRS]: 277-282
- Closed hashing [CLRS]: 269-271, 272-276
- Cuckoo hashing [MU]: 442-452 --- not covered this time
- Membership via Bloom filter [MU]: 114-118

More randomized dictionaries --- not covered this time

- Treap [MR]: 201-205, 206-207
- Skip list [MR]: 209-213

Priority queue

- Binary heap [CLRS]: 151-159, 162-165
- An appl of binary heap: heapsort [CLRS]: 159-161
- Height-biased leftist heap [HSA]: 424-428; [note]
- Binomial heap [CLRS1ed]: 400-416
- Fibonacci heap [CLRS]: 505-526

For disjoint sets

- Linked list and forest representations [CLRS]: 561-562, 564-572

Forest with union by rank and path compression [CLRS1ed]: 450-458

For graphs

Three representations [CLRS]: 589-592; [note]

Breadth-first traversal [CLRS]: 594-601

Depth-first traversal [CLRS]: 603-610

For trees --- not covered this time

Link/Cut tree [Tarj]: 59-70

Euler tour tree [wiki]

LCA queries via RMQ [NS]: 21-29

- [CLRS]: Introduction to Algorithms by Cormen, Leiserson, Riverst, and Stein, Third Edition.
- [HSA]: Fundamentals of Data Structures in C by Horowitz, Sahni, and S. Anderson-Freed, Second Edition.

- Additional resources are provided where necessary.

- Prereq denotes that this topic is typically taught in a prereq course.
- AR stands for additional reading (no lecture delivered but included in syllabus).
- EP stands for a problem of importance but it is given as part of an exam.
- NS says that it is not part of the syllabus although it was taught.

- The slides of C programming course (prereq) are accessible from here.