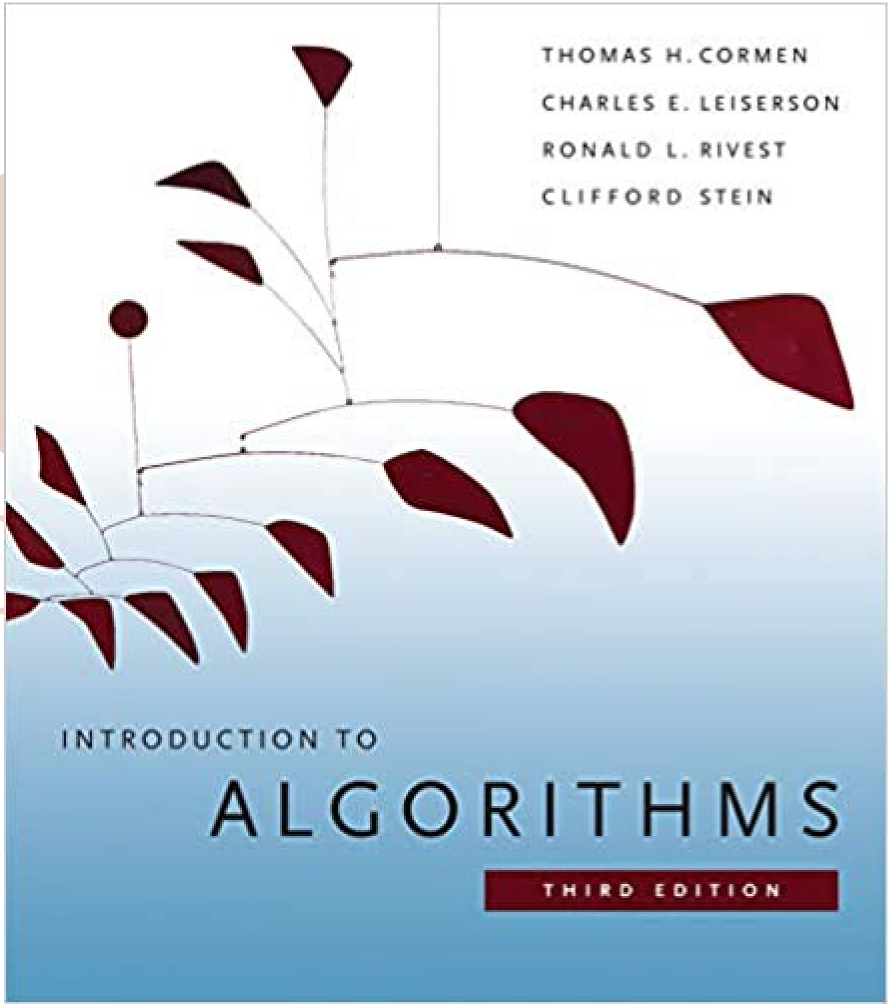
CS 07540 Advanced Design and Analysis of Algorithms

Finals Week

- Final Exam



Overview

Most problems will be blocks of four True/False questions, such as

Ⓣ Ⓕ FIND/SEARCH/GET in a BST with n nodes has runtime $O(\log(n))$.

with a few free response questions added. There will about 20 problems in total (the majority being blocks of four True/False). There will be no questions requiring you to program in Python or Java. However, standard algorithms we implemented in class or on homework may be asked for, such as

“Give a short description of an algorithm to find the minimum in a BST.”

Focus Points

- General terms and asymptotic notation
  - ADT and data structures
  - Meaning of O, Θ, Ω
- Heaps and Trees
  - Binary heaps as arrays and binary trees
  - Binary search tree structure

- Self-balancing trees and forests
    - Height-balanced trees
        - Red-Black tree structure
        - AVL tree structure
        - 2-3 trees structure
    - Weight/height-balanced trees
        - Scapegoat tree structure
    - Priority Queue
        - Binomial Heaps structure
        - Fibonacci Heaps structure

- Amortized Analysis and Lower Bounds
    - Run-time upper bounds for standard methods (Find, Insert, Delete)
        - Array
        - Dynamic array
        - BST (non-balanced!)
        - Scapegoat tree
        - 2-3 tree / Red-black tree
    - Run-time lower bound for comparison sort algorithms

- Algorithm Ideas
    - Binary search (in ordered array)
    - Search in BST (min/max)
    - Insertion Sort
    - Merge Sort
    - Heap Sort (with min or max heap)
    - Run-length Encoding
    - Huffman Trees
    - Predecessor/Successor in BST

- Runtime of Algorithms w/o Proof
  - Search in unordered array
  - Binary search (in ordered array)
  - Insertion Sort
  - Merge Sort
  - Heap Sort (with min or max heap)
  - FFT runtime (Fast Fourier Transform)
  - Fast/binary exponentiation