Final Overview

Analysis of Algorithms

Topics

- Week 8 Dynamic Programming
  - Solving via Dynamic Programming versus Divide and Conquer – 1 Problem
  - Shortest Path - 1 Problem
  - LCS – 1 Problem
- Week 9 Optimization
  - Linear Programming
  - Integer Programming
  - Formulation - Part of 2 Problems
- Week 10 Intro to NP – 1 Problem
  - Decision Problems versus Optimization Problems
  - Clique, Coloring, Hamiltonian Cycle, Vertex Cover, Sat
  - Regular Expression, Language, NFA, DFA
- Week 11 NP part 2 - 1 Problem
  - P, NP, Co-NP, NP-Complete language definitions
  - Verification
  - Reducibility

Topics

- Week 12
  - String Matching - 1 Problem
    - Calculate pi, and use it during KMP to find matches,
  - Binary Search Tree - 1 Problem
    - **Preorder, inorder, postorder, rotations**
- Week 13
  - Approximation Algorithms & Greedy – Part of 3 Problems
  - Max Flow & Min Cut
    - Ford-Fulkerson – 1 Problem
- Bonus FFT – Look at Figure 30.1 pg 904 in the book
  - fast way to multiply two nth degree polynomials
- Exam Wednesday December 9th 8-10 pm.
- Pick 6 out of 10 problems

Test Questions

- There will be questions on every main topic.
- Question difficulty will be between the quizzes and the homework.
- Pick 6 of 10 (plus bonus FFT).
- Know what a vertex cover is!

Grade

- In class work 10%
  - I drop one quiz
- Midterm 25%
- Final 25%
- Homework 40%
  - I drop one homework.

Outcomes - Qual

- Ability to analyze the complexity of an algorithm with techniques such as Masters Method, Amortized Analysis, and Decision Trees
- Ability to recognize and analyze several sorting algorithms including merge sort, heap sort, binary sort, bucket sort, selection sort, bubble sort, and quick sort
- Ability to design and analyze algorithms to find shortest path, minimal spanning tree, string matching, LCS, Max Network Flow
- Prove correctness of an algorithm
- Design an algorithm using a variety of techniques including: divide and conquer, dynamic programming, greedy methods, approximation methods, and Depth-First-Search .
- Analyze insert, delete, and search operations on advanced data structures such as Binominal heaps, Fibonacci heaps, Binary Search Trees
- Recognize and analyze decision problems and reduce to known P or NP-complete problems

Examples

- **What are the two main steps of induction, and how do they relate to the three steps of “proof of correctness” of a loop?**
- **Is longest path NP Complete or P?**
- **Is shortest path NP Complete or P?**
- **What does dynamic programming gain over divide and conquer?**