CS 312: Algorithm Analysis

Lecture #40: Review

Announcements

- Projects #5 and #6 Graded and on BlackBoard
  - Please check your early bonus count and your late day budget on BlackBoard
  - Estimate your grade
- Project #7 Due Today!
  - No Late Days
- Complete the Course Evaluation for "1 HW"
  - In sec. 002 & 003, this will mean that you will receive extra credit in the amount of the average value of a homework asst.
- CS 312 Ultimate Frisbee
  - 4/19, this Wednesday, First Reading Day
  - 10am at Kiwanis Park
- Thought

Announcements II

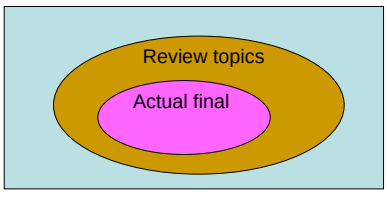
- Final Exam
  - Section 002:
    - Tuesday 4/25
    - 11am-2pm
    - Here
  - Section 003:
    - 4/25
    - 7pm-10pm
    - Here
- Students from sections 002 and 003 may elect to take exam with other section.
- Note: section 001 final is different!

Objectives

- To remind you of the topics we have covered and that you may encounter on the final.
- To get you thinking about synthesizing the various aspects of the course.

Review for Final

- 80% last 1/3 of class (since Mid-Term 2)
- 20% synthesizing the entire class



Topics (as posted on Blog)

- Definitions of Big-O, Big-Omega, Big-Theta
- How to prove that a problem is NP complete.
- Deriving the big-Theta bounds for certain kinds of divide-and-conquer algorithms using the methods for solving recurrence relations covered in class. In particular, be sure to be able to do the first problem in the last homework assignment on recurrences. This will involve knowing some logarithm and exponent identities.
- How backtracking algoirthms such as depth-first and breadth-first search work.
- The minimax algorithm.
- Design decisions in creating and implementing branch and bound algorithms.
- Basics of solving the Job Assignment and TSP problems with branch and bound algorithms.
- How to compute amplification of stochastic advantage for repeated Monte Carlo algorithms.
- How to express a Linear Programming problem in Slack Form in preparation for solution by the Simplex Method.
- How to compute speedup and efficiency for parallel algorithms and how to tell if a parallel algorithm is good or not.

Last 1/3 of CS 312

- Backtracking (DFS, BFS, etc.)
- Branch and Bound
- TSP
- Probabilistic Algorithms
- Linear Programming
- Parallel Algorithms

Backtracking

- Elements of a backtracking algorithm
  - DFS
  - BFS
  - …
- Compare with branch and bound
- Example:
  - 8 queens
  - Graph Coloring

Branch and Bound

- Elements of a branch and bound alg.
  - State Definition
  - State Expansion Strategy
  - Upper bounds (BSSF)
  - Lower bound – via Bounding Function
  - Pruning

Probabilistic Algorithms

- Las Vegas
  - E.g., 8 queens
- Monte Carlo
  - E.g., Primality testing
  - Amplification of Stochastic Advantage
- Numerical Probabilistic
  - E.g., Buffon's needle

Linear Programming

- Formulating Problems
  - Objective Function
  - Constraints
- Standard Form
- Slack Form
- Concise Representation of Slack Form
- Simplex Algorithm

Parallel Algorithms

- Speedup
  - Why is this important?
  - What makes a good parallel algorithm?
- Efficiency

Big Picture & Synthesis

- Definitions of Big-O, Big-Omega, Big-Theta
- How to prove that a problem is NP complete.
- Derive the big-Theta bounds divide-and-conquer algorithms using the methods for solving recurrence relations covered in class.
  - Work an example (merge sort)
- Compare and contrast all major paradigms
  - Greedy
  - Divide and Conquer
  - Dynamic programming
  - Backtracking
  - B&B
  - Probabilistic
- Design and analyze algorithms for problems you've never seen
- Explain empirical vs. theoretical analysis
  - Think of your experience with TSP
  - Remember randomized pivot selection for Quicksort

Good Luck!