Class Summary

Slides by Carl Kingsford

May 2, 2014

Topics

Minimum spanning tree (Prim's, Kruskal's, Reverse Delete); Heaps, Union-Find (tree-based and array-based); Clustering, Graphs; Traveling Salesman (NP-hardness, 2-approximation); DFS; BFS; Topological sort; bipartite testing (aka 2 coloring); Dijkstra's algorithm; A*; Bellman-Ford; Counting Inversions; Closest Points in the Plane (divide and conquer + randomized); Matrix multiplication; fast integer multiplication; Binary search trees; Splay Trees; Subset Sum; String Edit Distance (normal and linear-space); Segmented least squares; Matrix Chain Multiplication; Optimal binary search trees; Suffix trees (use and construction); Network flow; Maxflow=Mincut; Matching; Image segmentation; circulations with demands; airline scheduling; linear programming; integer programming; P, NP; NP-completeness (3SAT, Hamiltonian Path, TSP, Graph Coloring, three-dimensional matching); Metropolis algorithm; Randomized global mincut via contractions.

Algorithm Design Techniques

1. Greedy
2. Tree-growing
3. A*
4. Divide and conquer
5. Dynamic Programming
6. Network Flow
7. Linear and Integer Programming
8. Reductions (and NP-completeness)
9. Randomized algorithms

Data Structures

1. Heaps
2. Union Find
3. Graphs
4. Binary Search Trees
5. Splay trees
6. Suffix Trees

Final Exam

- The final will cover the entire course.
- Possible topics: Slides, relevant book chapters, anything covered in class, homeworks, midterms.
- There will likely be some bias toward topics covered since the 2nd midterm.
- You will have 3 hours for the final, but the final will be designed to take less than that.
- The final will be in Porter Hall room 100.

Design of the Course

- “I wish there were more programming assignments.”

- “We should do more examples in class. We should also talk more about how these algorithms are used in practice.”

- “We didn’t cover algorithm X.”

Design of the Course

- “I wish there were more programming assignments.”
  A lot of comments amount to: “I’d like to do more assignments of a style that I am already very comfortable with.”

- “We should do more examples in class. We should also talk more about how these algorithms are used in practice.”
  The course is about abstract thinking. One of the goals is to avoid having to work out specific examples to understand a concept. Single examples can be very difficult and also require a lot of synthesis to predict how those examples generalize.

- “We didn’t cover algorithm X.”
  Our goal was to teach **how to learn** about algorithms, not specific algorithms, which will come and go in importance.

Course Evaluations

If you enjoyed the course, please fill out the course evaluation:

http://www.cmu.edu/hub/fce/

Thanks!