CSE 326
Lecture 23

Spring Quarter 1997

Summary of the the Course

- Lists, Stacks, and Queues
  - Basic multipurpose data structures
  - Managing a private store
  - Cursor implementation of pointers
  - Applications
    - > polynomials
    - > text editing *
    - > adjacency list representation of a graph and depth first search (Section 9.1.1, page 320 and sections 9.6.1 and 9.6.4, pages 357 and 366)
    - > sparse matrices *

* indicates that the topic was not covered thoroughly in the book.

Summary Continued

- Trees
  - Pointer representation of multi-way trees
  - Tree traversals
    - > preorder
    - > postorder
    - > inorder
  - Simple binary search
  - Binary search trees
    - > Generic binary search trees
    - > AVL trees
    - > Splay trees
  - B-trees
  - Multidimensional search trees
    - > Quad trees *
    - > k-d trees *

Summary Continued

- Hashing
  - Choosing a hash function
    - > division method
    - > mulitplication method *
    - > universal hash functions *
    - > hashing strings and general structures
  - Open hashing (chaining)
  - Closed hashing
    - > linear probing
    - > quadratic probing
    - > double hashing
  - Rehashing
  - Extendible hashing

Summary Continued

- Priority queues
  - Binary heap
  - d-heap *
  - Leftist trees
  - Skew heaps
  - Calendar Queue *
  - Applications
    - > Discrete event simulation *
    - > N-body problem *
- Sorting
  - Insertion sort
  - Quicksort
  - Mergesort
  - Heapsort
  - Radix sort

Summary Continued

- Dynamic equivalence
  - Up-tree implementation
  - Weighted Union
  - Path Compression
  - Applications
    - > Minimum spanning tree, Kruskal's algorithm (Section 9.5.2 page 354)

Summary Continued

- Programming Methods
  - Recursion
  - Call by reference vs. call by value
  - Memoization
  - Dynamic programming (Section 10.3, page 416)
  - Converting a recursive algorithm to an iterative one.
    - > Eliminating end recursion
    - > Using a stack
    - > Observing the recursion as a two-pass iteration, as an example the AVL tree insertion or deletion.
  - Balancing to achieve good performance (AVL, weighted union)
  - Divide and conquer (mergesort, fast polynomial multiplication, binary search, ...)
  - Looking for good amortized performance (Splay, Skew, Rehash)
  - Understanding principles of good cache performance
  - General familiarity with more advanced data structures and their applications.

Summary Concluded

- Analysis
  - Analyzing basic loops
  - Developing and solving recurrences which arise when using recursion.
  - Upper bound vs. lower bound analysis
  - Worst case vs. average case analysis
  - Amortized analysis vs. worst case per operation analysis
  - Correctness proofs
- Problem solving practicum
  - Most homework involved problems solving rather that rote practice. Problem solving skills are among the most valuable skills you can have.
- Programming practicum
  - More familiarty with object oriented programming
  - Strengthening of C++ experience.
  - Introduction to Unix
  - Programs required a combination of problems solving and coding known solutions. This is what many of you will do in your future employment.

Text Coverage

- Chapters 1-5 and 8 all sections
- Chapter 6 all except section 6.8
- Chapter 7 all except sections 7.3-4, 7.8-7.11
- Also parts of:

  - Section 9.5.2 Kruskal's algorithm
  - Section 9.1.1 Representation of graphs
  - Sections 9.6.1 and 9.6.4 Depth-first search

Topics not Covered in the Book

- Sparse matrices
- Text editing data structures
- Multiplication method hash function and universal hash functions
- Calendar queue
- Cache performance

Final Exam

- Wednesday, June 11, 2:30 - 4:20
- Exam starts promptly at 2:30 and all papers must be turned in by 4:20.
- Sit in alternate seats.
- Exam covers all the material in the course.
- Must use 8.5 x 11 inch blue book.
- You may use part of your blue book to prepare your own handwritten notes.
- No other notes or books may be used during the exam. Calculators can be used but are not required. There may be some simple calculations to do, but they can be done by hand.
- Content of the Exam:

  1. Understand the methods and algorithms well. Drill yourself this.
  2. Problem solving skill. The homeworks were practice for this.