CSE 326: Data Structures Final Exam Review

Kathleen Tuite

Winter 2008

Last Lecture

Announcements

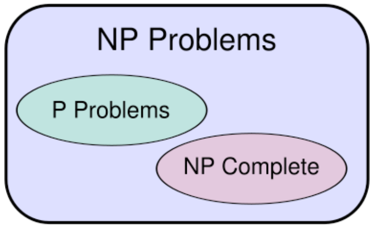
- Exam Tuesday 2:30 pm, here
  - Logistics: same as midterm (closed book, but 1/2 page of new handwritten notes, plus the notes you had for the midterm)
- Office hours/review next week
  - Monday: Monday, CSE 503, 4:30-5:30++
- HW return, etc.
  - Most hw returned by today; with luck the last one will be ready on Monday afternoon
  - Project 3 grades by the end of finals week

Our ADTS (& implementations)

- Stack (array, list)
- Queue (array, list)
- PQ (various flavors of heaps)
- Dictionary (various flavors of trees, hash tables)

Other stuff we learned about

- Asymptotic Analysis
- Union Find
- Sorting
- Graphs
  - Searching
  - Topological sort
  - Shortest path
  - MST
- "Cultural" Topics (i.e. you don't need to understand the finer details for the test):
  - Dynamic programming
  - NP

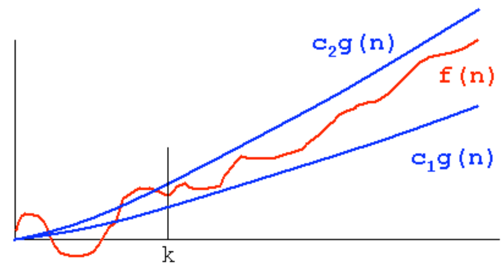


Pre-Midterm Topics (1)
(A more thorough list…)

- Linked lists. Simple linked lists, doubly linked lists, circularly linked lists.
- Stacks and Queues, array and list implementations.
- Recursion. Designing algorithms recursively.
- Asymptotic analysis, Big-O. Worst case, upper bound, lower bound, analyzing loops, recurrences, amortized complexity.
- Trees – definitions

Asymptotic Analysis

- Big-O: upper bound
- Big-Theta: tight bound (both O and Ω)
- Big-Ω: lower bound



Recurrence Relations

```
mergesort(array, left, right){
   if (right - left >1){
        mid = floor((right - left)/2);
        mergesort(array, left, mid);
        mergesort(array, mid+1, right);
        merge(array, left, mid, right);
   }
}
```

Running time: **$T(n) = 2\,T(n/2) + n$** = $O(n \log n)$

Trees

- Binary tree: root, left subtree (possibly empty), right subtree (possibly empty)
- Tree nodes: data, left child ptr, right child ptr
- For a tree of height h:
  - Max # of leaves: 2^h
  - Max # of nodes: 2^{h+1} -1
  - Min # of leaves: 1
  - Min # of nodes: h+1
- Inorder/preorder/postorder traversal

Pre-Midterm Topics (2)

- Priority queues – definition and operations.
- Binary Heaps, D-heaps - Findmin, Deletemin, Insert. Additional operations of increase, decrease, buildheap.
- Leftist Heaps and Skew Heaps - Findmin, Deletemin, Insert. Additional operations of merge, increase, decrease
- Binomial Queues - Findmin, Deletemin, Insert. Additional operations of merge, increase, decrease.
- Dictionary ADT
- Binary search trees – Inorder, preorder, postorder traversals, insert, delete, find.
- AVL trees - Single and double rotations, insert, find.

Priority Queues

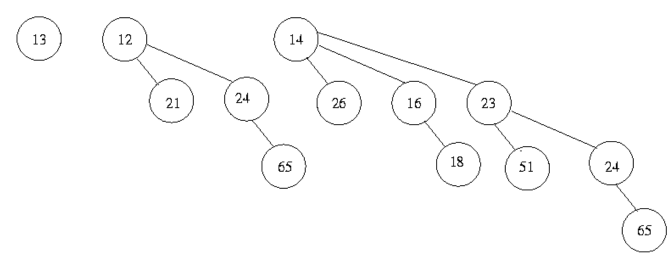
- Operations
  - Find min
  - Delete min
  - Insert
  - Change priority
- Applications
  - Scheduling, heap sort, greedy algorithms where you always want to get the next smallest/largest thing

Heaps

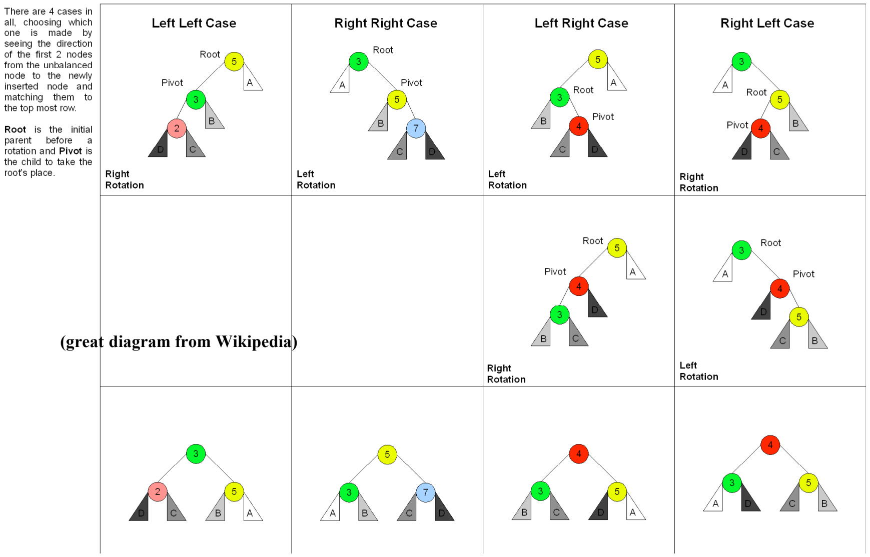
- Heap property (min heap): node value is always less than the value of its children
- Leftist Heap
  - Leftist tree + heap property
  - For every node x, npl(left(x)) >= npl(right(x))
  - Result: tree tends to be left-heavy
  - Every subtree of a leftist tree is leftist
  - Special merge, increase, decrease operations
- Skew Heap
  - Like Leftist Heap but always swap

More Heaps

- Binomial Queues
  - Forest of trees
  - Min could be any tree's root



AVL Trees

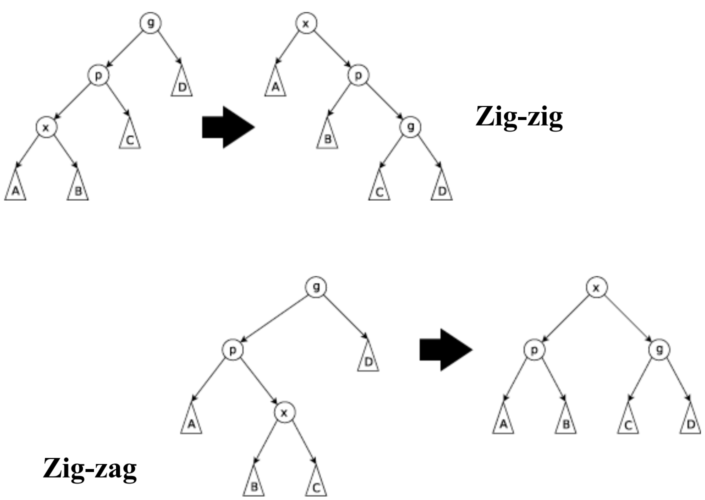


(great diagram from Wikipedia)

Post Midterm Topics (1)

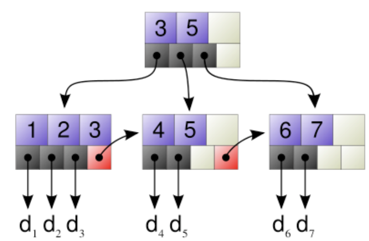
- Splay trees - Splaying, insert, find.
- B-trees. Motivation (esp. large data on secondary storage), choice of M and L, insert (no delete).
- Hashing. Properties of good hash functions. Selecting hash table size. Separate chaining and open addressing. Linear Probing, Quadratic Probing, & Double Hashing to resolve collisions. Rehashing.
- Disjoint Union/Find. Up-trees. Weighted union (union by size) and path compression.
- Sorting. Insertion sort, Selection sort, Heap sort, Merge sort, quicksort.
- Bucket sort, Radix sort. Lower bound on comparison sorting. In-place sorting. Stable sorting.

Splay Trees



B-Trees

- Block-oriented storage, size of each node = size of 1 page on disk
- M: number of children of interior nodes (M-1 keys)
- L: number of data values (including keys) in each leaf node
- Remind me to do an example on the board at the end of class



Hashing

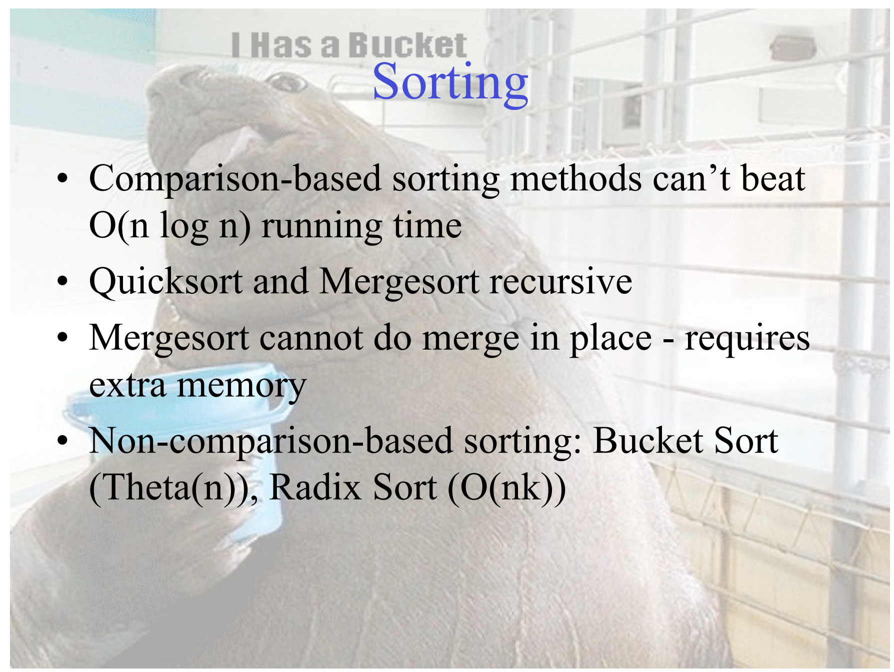
- Separate chaining (lambda can be > 1)
- Closed hashing
  - Linear and quadratic probing, double hashing
  - Lambda must < 1, < .5 for quadratic
- Importance of a good hash function

When is closed hashing better than separate chaining?

- Not very often…
- But, for small record sizes (a few words or less) the benefits are:
  - More space-efficient since no pointers or need to allocate extra space
  - More time-efficient since no need to allocate extra space
  - Better locality
  - Easier to serialize
- Good for portable devices with small memory/processing power
- Good for multithreaded use

Union Find

- To do a union on two nodes that aren't root nodes… do a find on both of them first to *get* the root nodes (set representatives) and union the roots
- If nodes are already roots, find is fast
- If not, you get your trees compressed along the way



Sorting

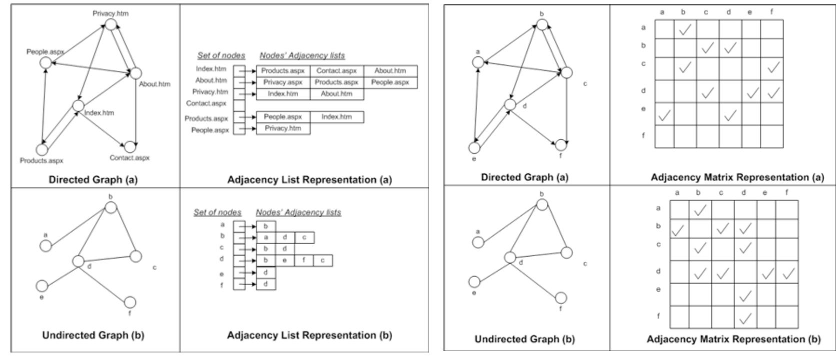
- Comparison-based sorting methods can't beat O(n log n) running time
- Quicksort and Mergesort recursive
- Mergesort cannot do merge in place - requires extra memory
- Non-comparison-based sorting: Bucket Sort (Theta(n)), Radix Sort (O(nk))

Post Midterm Topics (2)

- Graphs. Directed and undirected. Adjacency list and adjacency matrix representations.
- Topological sorting.
- Graph searching. Depth-first, breadth-first search, best-first search.
- Shortest paths. Dijkstra's algorithm. Greedy Algorithms.
- Minimum spanning tree: Prim's and Kruskal's algorithms
- Dynamic programming: Floyd-Warshall shortest-paths algorithm

Graphs

- List of vertices and edges
- Adjacency list, adjacency matrix



Graph Alg Running Times

Algorithm	*Running Time*
DFS/BFS	O(|V| + |E|)
Topological Sort	O(|V| + |E|)
Dijkstra	O(|E| log |V|)
Prim	O(|E| log |V|)
Kruskal	O(|E| log |E|)

And That's it…

- Hope you've learned a lot
- Good luck on the final & best wishes for the future
- Time for whiteboard examples / questions?