Course Review for Midterm II

Cpt S 223

Fall 2009

Midterm Exam 2

- <u>When:</u> Friday (11/20) 10:10 -11am
- <u>Where:</u> in class

- Closed book, closed notes
- Comprehensive
  - With primary focus on the following topics: priority queues, hashing and disjoint sets.
  - Other topics covered before Midterm exam 1 could be included although indirectly.

- Material for preparation:
  - Lecture slides & in-class notes
  - Homeworks & program assignments
  - Weiss book

Priority Queues

- Binary heap, Binomial heaps
- Definitions:
  - Binary heap
    - Structure property:
      - complete binary tree except the last level (filled breadth-first left to right)
    - Heap order property
      - Min heap : each node's value is less than or equal to its children
  - Binomial heap
    - Structure property:
      - A forest of binomial trees (similar to binary representation)
    - Heap order property:
      - Min heap: within each binomial tree, same heap order like in binary heap

Implementation

- Binary heap
  - Tree structure can be implemented as an array
    - Where nodes are stored in breadth-first order
    - Children of node at A[i] are at: A[2i] and A[2i+1]
    - Conversion procedure: Array to tree, tree to array
- Binomial heap
  - Array of pointers to each binomial tree
    - log n binomial tree pointers
  - Know relationships between binomial heap, binomial trees and their properties

Run-times for each heap operation

Two main techniques: percolateUp and percolateDown

	Insert	DeleteMin	Merge
Binary heap	O(lg n) worst-case, O(1) using buildheap	O(lg n)	O(n)
Binomial Heap	O(lg n) worst-case O(1) for insertion in sequence	O(lg n)	O(lg n)

Other operations:

- deleteMax()
- decreaseKey(p,v), increaseKey(p,v)
- remove(p)

Union-Find data structure

- Union-find
  - Supports two operations on disjoint sets:
    - Union(a,b)
    - Find(a)
- One application is equivalence class computation:
  - Maximal subsets defined by equivalence relation
  - Disjoint subsets
- Array implementation

Steps in the Union (x, y)

Steps in Union(x,y):

1. $EqClass_x$ = Find (x)
2. $EqClass_y$ = Find (y)
3. $EqClass_{xy}$ = UnionSets ($EqClass_x$, $EqClass_y$)

Equivalence class algorithm:

- Initially, put each element in a set of its own
- FOR EACH element pair (a,b):
  - Check [a R b = true]
  - IF a R b THEN
    - Union(a,b)

Variations in Union

- Simple Union (aka Arbitrary Union)
  - Perform arbitrary linking of roots
- Smart Unions
  1. Union-by-Rank
     - Connect shorter tree under taller tree
  2. Union-by-Size
     - Connect smaller size tree under larger size tree (size(tree) = #nodes in the tree)

Variations of Find

- Simple Find
  - Simply returns the root id without modifying the tree
- Smart Find
  - Uses path compression
    - Link all nodes along the path from x to the root directly to the root
  - Returns the root id, but the resulting tree could have been modified as a result of path compression

Heuristics & their Gains

	Worst-case run-time for m operations
Arbitrary Union, Simple Find	O(m n)
Union-by-size, Simple Find	O(m log n)
Union-by-rank, Simple Find	O(m log n)
Arbitrary Union, Path compression Find	O(m log n) Extremely slow Growing function
Union-by-rank, Path compression Find	**O(m Inv.Ackermann(m,n)) = O(m log*n)**

Hashing

- Hash functions
  - purpose: string to integer, integer to integer
- Choice of a “good” hash functions
  - Reduce chance of collision
  - Relatively smaller key value
  - Does not need huge hash table size
- Hash Tables use hash functions
- Hash table size should be a prime
- Load factor
  - Measure to tell how crowded a hash table is
- Know algorithms & analysis for the following
  - Collision resolution by chaining
  - Collision resolution by open-addressing
    - Linear probing, quadratic probing
    - Double hashing
  - Rehashing
- Think of applications where hash tables could work out better than other data structures.

General Tips

- Work out examples of basic operations
- Show steps whenever possible
- Questions mostly objective, but be prepared to see one or two subjective ones too
- Write less and to the point
- Don’t leave questions totally unanswered
  - If unable to solve, write at least your approach idea(s)