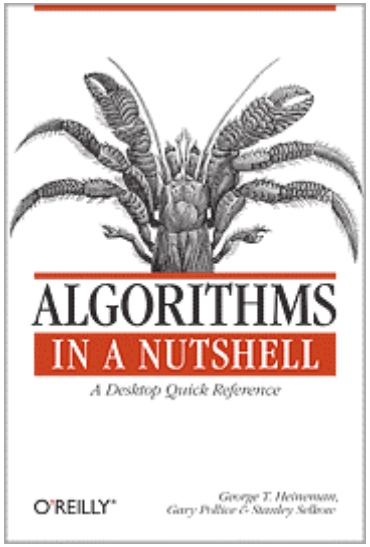
Algorithms in a Nutshell



Session 10

Summary

4:20 – 4:45

Outline

- Lessons Learned
- When All Else Fails
- Summary

Principles of Algorithm Design

- Know your data
- Decompose problem into smaller problems
- Choose right data structures
- Add storage to increase performance
- If no solution is evident
  - Construct a search
  - Reduce problem to another problem with solution
- Testing algorithm implementation is hard!

Know your Data

- Normal distribution of data
  - Surprising efficiency of BUCKET SORT and HASH SORT
  - Dense points in Convex Hull removed by Akl-Toussaint heuristic
  - *d*-dimensional (negative) impact on Nearest Neighbor Query

Problem Decomposition

- Divide and conquer
  - Solve smaller version of same problem
  - BINARY SEARCH
- Solve problem with different sub-problems
  - The use of `partition` within QUICKSORT
  - Build partial hulls for CONVEX HULL SCAN
  - Use of `heapify` in HEAP SORT

Choose Right Data Structures

- Use of Priority Queue in LINE SWEEP
  - Don’t use default priority queue implementation!
  - Wrong structure leads to O(n) on key operations
- Graph Storage
  - Sparse vs. Dense Graphs
  - Adjacency matrix vs. adjacency lists

Add Storage for Performance

- Cache computations that won't change
  - `hashCode()` in java.lang.String class
- Use associative index to lookup values
  - Rather than search within list
- Priority Queue for DIJKSTRA'S ALGORITHM
  - Extra storage to enable decreaseKey operation
- General device to prevent search state explosion
  - Closed state sets

Reduce Problem to Another Problem

- Network Flow Algorithms (Chapter 8)
  - FORD-FULKERSON Family of algorithms

Testing Algorithms is Hard

- Compare against Brute Force, if available
- Floating point computations quite challenging
- Use Visualization to aid understanding
  - More debugging than testing
  - Useful for comparing against past (textbook) solutions

When All Else Fails

- Consider relaxing four key assumptions
  - Answers must be exact
  - One instance of a problem is being solved at a time
  - Sequential computing platform
  - Deterministic computing platform

Approximation Algorithms

- Decrease time to return an answer
  - You may have useful bounds as to how close answer is to actual answer
- Traveling Salesman Problem (TSP)
  - In 1976, major milestone by Christofides
  - Efficiently computes tour that is no more than 50% longer than the shortest tour

Offline Algorithms

- Searching assume random likelihood of sought-for target
  - What if the set of targets is known in advance?
  - And available all-together?
  - Useful optimizations possible

Parallel Algorithms

- Entire books have been written on this subject
- Most algorithms have parallel counterparts

Probabilistic Algorithms

- Size Estimation
  - Run repeated random trials
  - Over time, approach accurate estimate
- Algorithms that can return wrong answer with diminishing probability
  - Repeated execution leads to increased "confidence" in an answer
  - Miller–Rabin primality test

Final Slide

- Thank you for your patience today
- Feel free to send me follow-up questions