Course Summary

1. **Introduction**
2. **Tree Data Structures**
3. **Average-Case Analysis**
4. **Amortized Analysis**
5. **Randomization/Expected Case**
6. **Online/Competitive Analysis**
7. **Complexity Theory Intermission**
8. **Approximation**
9. **Parallel Time**
10. **Memory**

each chapter as *appetizer/door opener* into its own full-fledged contemporary research direction…

Course Overview

Design and Analysis of Algorithms

with respect to various notions of performance:

Modes of Analysis:

- worst-case,
- average-case,
- expected-case,
- amortized;
- competitive ratio,
- approxim.ratio etc.

Cost Measures:

(sequential) time

parallel runtime

memory use

#processors/gates

(communication volume)

etc …

Purpose **of Algorithm Analysis**

Tradeoffs: *polynomial* vs. $O(\ldots)$ vs. constant factors;

Count #operations, bit-cost, w/o access latency etc.

Goal: Predict the behavior of an algorithm

- before actual execution
- independent of hardware details (e.g. clockrate)
- realistically but simple
- Indicate im/possibility of improvement/optimality.

Cost Measures:

(sequential) time

parallel runtime

memory use

#processors/gates

(communication volume)

etc …

§1 Introduction & Recap

- "Virtues" of Computer *Science*
- Power of Abstraction
- Importance of asymptotic efficiency
- Classes of asymptotic growth
- Sorting: specification and optimality

§2 Tree Data Structures

- Abstract Data Types
  - Hide hardware/implementation/data structure
  - Recap: basic / derived/ linked data structures
- AVL Trees:
  - definition, properties
  - operations/maintenance, cost, deficiency
- Binomial Trees, Binomial Heaps:
  - definition, operations, analysis
  - ExtractMin, DecreaseKey, **Merge** in $O(\log n)$

§3 Average-Case Analysis

- Purpose&Modes of Algorithm Analysis
- Motivation: Incrementing a Binary Counter
- Example: naïve QuickSort
- Simplex Algorithm
- *Smoothed* Analysis

pivot choice

§4 Amortized Analysis

- Algorithmic Cost Analysis
  - Motivation Bit-Cost of Repeated Increment/Binary Add.
  - Amortized vs. Average vs. Worst-Case Analysis
- Fibonacci Heaps:
  - Relaxed Binomial Trees
  - ExtractMin and DecreaseKey
- Minimum Spanning Tree
  - Prim's Algorithm with Binomial vs. Fibonacci Heap
  - Kruskal's Algorithm: *Union-Find* data type
- Disjoint Set Data Structures
  - Fast and Slowly Growing Functions
  - Analysis of *Union-by-Weight*
  - Lazy Union-by-Rank with Path Compression

§5 Randomization

- Motivation: Reliability
- Sources of Randomness
- Las Vegas vs. Monte Carlo
- Primality Testing
- Errors and Amplification
- Blackbox Polynomial Test
- Schwartz-Zippel Lemma
- Perfect Matchings in Graphs
- Matchings via Tutte Determinant

§6 Competitive Analysis of Online Algorithms

- Motivation: Ski Rental
  - *Break-Even* Algorithm
  - is 2-competitive; optimality
- Online Paging
  - *Least-Recently Used* is k-competitive
  - *Least-Frequently Used* is not competitive
  - LRU is optimal among deterministic online
- Randomization and expected competitiveness
  - 1.84-competitive randomized Ski Rental

§7 Complexity Theory

- Complexity Classes $\mathcal{P}$ and $\mathcal{NP}$
- Eulerian/Hamiltonian Cycle
- Edge/Vertex Cover
- Clique, Independent Set
- Comparing Decision Problems
- Travelling Salesperson (TSP)
- Knapsack

§8 Approximation

- metric Travelling Salesperson
  - *Christofides*: approximation ratio 2
- Knapsack
  - Strongly polyn.-time / Dynamic Programming
  - *Fully Polynomial-Time Approximation Scheme*
- Limits of Approximability

§9 Fast Parallel Algorithms

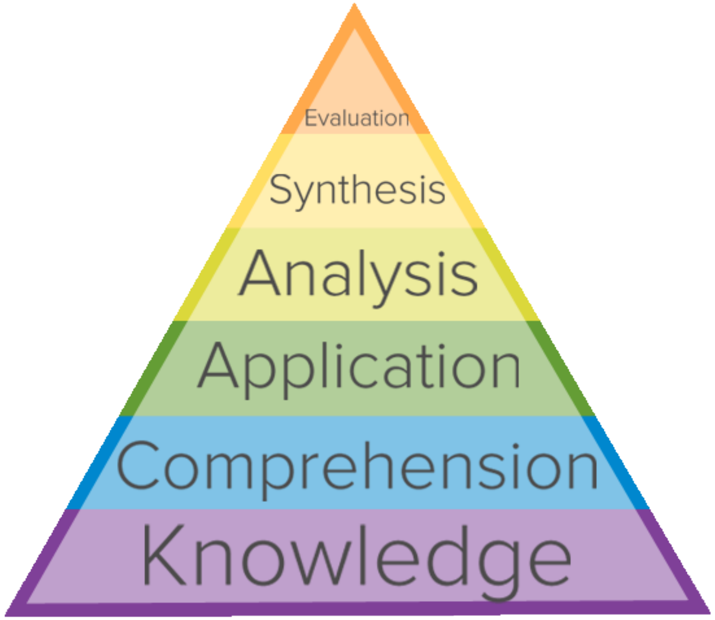
- Classification
- Parallel Prefix
- Carry-Lookahead
- Parallel Matrix
- Sorting Networks

§10 Memory-Efficient Algorithms

- Motivation
- Fibonacci, revisited
- Concise Boolean Matrix Powering
- Graph Reachability, revisited
- Cost/Methods of Memory Saving
- Memory ≈ Parallel Time
- Streaming Algorithms
- Memory hierarchy-aware algorithms

Pedagogy/Teaching Philosophy

Bloom's Hierarchy
of cognitive learning:



Konrad Lorenz:
(Nobel Prize 1973)

- *What is thought is not said*
- *What is said is not heard*
- *What is heard is not understood*
- *What is understood is not believed*
- *What is believed is not yet advocated*
- *What is advocated is not yet acted on*
- *What is acted on is not yet completed*