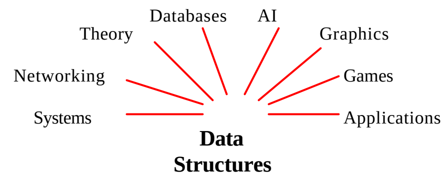
CSE 326: Data Structures and Algorithms
EOQ Wrap-up and Review

Hannah Tang and Brian Tjaden
Summer Quarter 2002

Why (did we) study data structures?

Clever ways to organize information in order to enable efficient computation



Why (did we) study algorithms?

Clever ways to solve problems in a *structured* and *efficient* manner

Ways to measure *structure*:
- Proofs of correctness
- Intuition

Ways to measure *complexity*:
- Worst case
- Expected case
- Average case
- Amortized over a series of (presumably representative) runs
- Best case (occasionally useful)

Apocalyptic Laptop

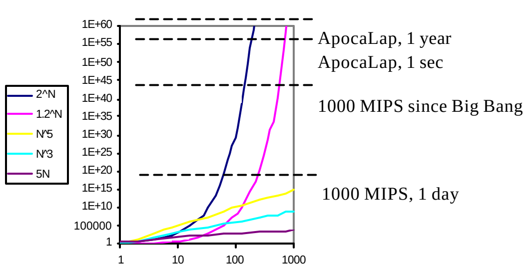
Seth Lloyd, a physicist at the Massachusetts Institute of Technology, has calculated how to make PCs almost unimaginably faster--if you don't mind working on a black hole.

Lloyd has used the laws of thermodynamics, information, relativity, and quantum mechanics to figure out the ultimate physical limits on the speed of a computer…

Lloyd's ultimate laptop would convert all of its 1-kilogram mass into energy via Einstein's famous equation $E = mc^2$, thus turning itself into a billion-degree blob of plasma. "This would present a packaging problem," Lloyd admits … The computer would be capable of performing 10^{51} operations per second.

- Charles Seife, Science Magazine, Vol289, No 5484, Sep 1 2000, pp. 1447-1448

Big Bang



Why (did we) take CSE 326?

- Learn some of the fundamental data structures and algorithms in computer science
  - And understand their tradeoffs!
- Learn to see and solve problems abstractly
  - Be able to see the intrinsic problem behind real-world scenarios, or vice versa, be able to realize an abstract solution in the real world
  - Data structures are your problem-solving building blocks!
- Learn to analyze and improve algorithms
  - Prove correctness
  - Gauge and improve time complexity
- Become modestly skilled with the UNIX operating system
- Appreciate that all languages are not created equal...

Some Data Structures, Algorithms, and Techniques We Covered

- List ADT
  - Stack ADT
  - Queue ADT
- Collection/Set ADT
- Dictionary/Map ADT
- Priority Queue ADT
- Disjoint Sets ADT
- Graph ADT

- Asymptotic analysis techniques
- Comparison-based sorting algorithms
- Shortest-path algorithms
- Minimum-spanning tree algorithms
- Randomization techniques
- “Greedy” algorithmic techniques
- Divide and Conquer techniques
- Dynamic programming techniques

Why So Many Data Structures?

Ideal data structure:

“fast”, “elegant”, memory efficient

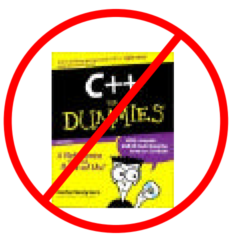
Generates tensions:

- time *vs.* space
- performance *vs.* elegance
- generality *vs.* simplicity
- one operation’s performance *vs.* another’s

The study of data structures is the study of tradeoffs. That’s why we have so many of them!

Are we convinced yet?

Mastery of this material separates you from …



Final Review: What you need to know

- Basic Math
  - Logs, exponents, summation of series
  - Proof by induction

$$\sum_{i=1}^{N} i = \frac{N(N+1)}{2}$$

$$\sum_{i=0}^{N} A^i = \frac{A^{N+1}-1}{A-1}$$

- Asymptotic Analysis
  - Big-oh, Theta and Omega
  - Know the definitions and how to show f(N) is big-O/Theta/Omega of (g(N))
  - How to estimate Running Time of code fragments
    - *E.g.* nested “for” loops
- Recurrence Relations
  - Deriving recurrence relation for run time of a recursive function

Final Review: What you need to know

- Lists, Stacks, Queues

Final Review: What you need to know

- (Binary) Search Trees

Final Review: What you need to know

- Hashing

Final Review: What you need to know

- Priority Queues

Final Review: What you need to know

- Sorting Algorithms

Final Review: What you need to know

- Graph Algorithms

Final Review: What you need to know

- Algorithm Design Techniques

Final Review: What you need to know

- Disjoint Sets

- Multi-dimensional search structures

Top Four Things to Know in Computer Science*

3) Ask not what you can do to an object; ask what an object can do to itself

2) $1 + 2 + \ldots + n = n(n+1) / 2$

1) The answer is "it depends"

0) $2^{10} = 1024$

** By Owen Astrachan, of Duke University*
Courtesy of Steve Wolfman