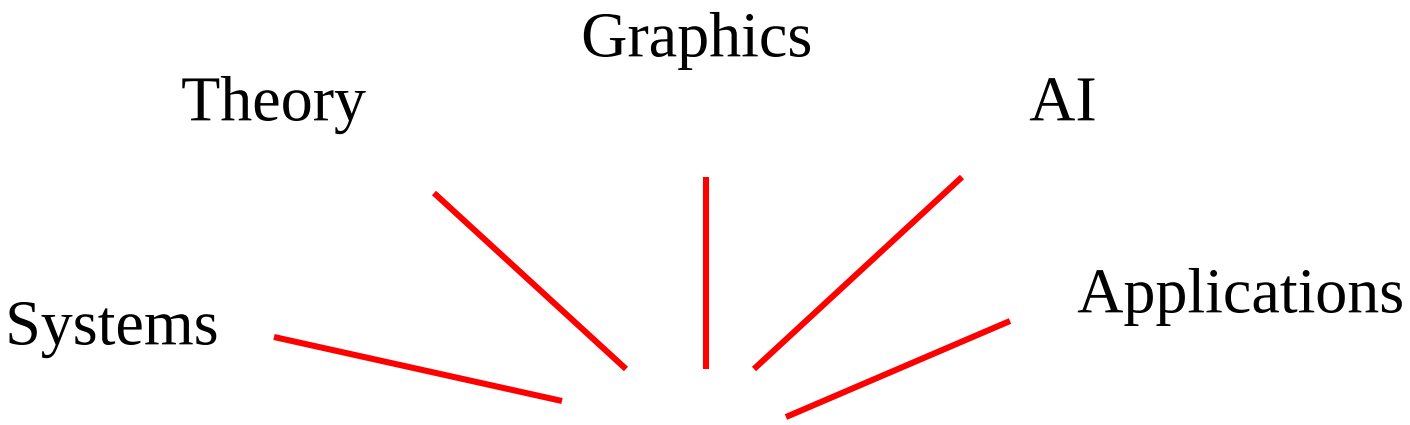
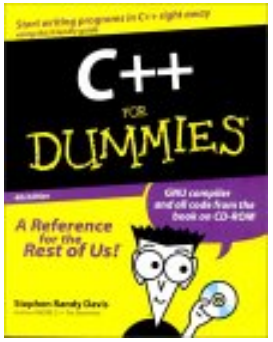
Introduction to Data Structure



Used Everywhere!

Mastery of this material separates you from:



- ***Perhaps the most important course in your CS curriculum!***
- ***Guaranteed non-obsolescence!***

What is this Course About?

Clever ways to organize information in order to enable efficient computation

Clever ways to organize information

- Lists, Stacks, Queues
- Heaps
- Binary Search Trees
- AVL Trees
- Hash Tables
- Graphs
- Disjoint Sets

Data Structures

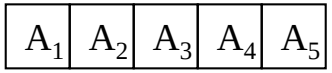
Efficient computation

- Insert
- Delete
- Find
- Merge
- Shortest Paths
- Union

Algorithms

Example

- Array data structure



- Algorithm to find the index of element x

```
for(int i=0; i<n; i++)
    if(a[i]==x)
        return i;
```

Efficient Computation

Our notion of efficiency:

How the running time of an algorithm scales with the size of its input

- several ways to further refine:
  - worst case
  - average case
  - amortized over a series of runs

Asymptotic Complexity

- Example
  - Access the k-th element in an array
    - O(1)
  - Searching for an element in a sorted array
    - Exhaustive search: O(n)
    - Binary search: O(log n)
  - Sort an array
    - Bubble sort: $O(n^2)$
    - Quicksort: O(n log n) on average

Asymptotic Complexity

- This course is about asymptotic complexity
  - Deals with algorithms, and their analysis
- It is *not* about better coding
  - Does not deal with coding details

I don't need to
take CS101 because:
• I can buy a faster laptop
• I can write very good code

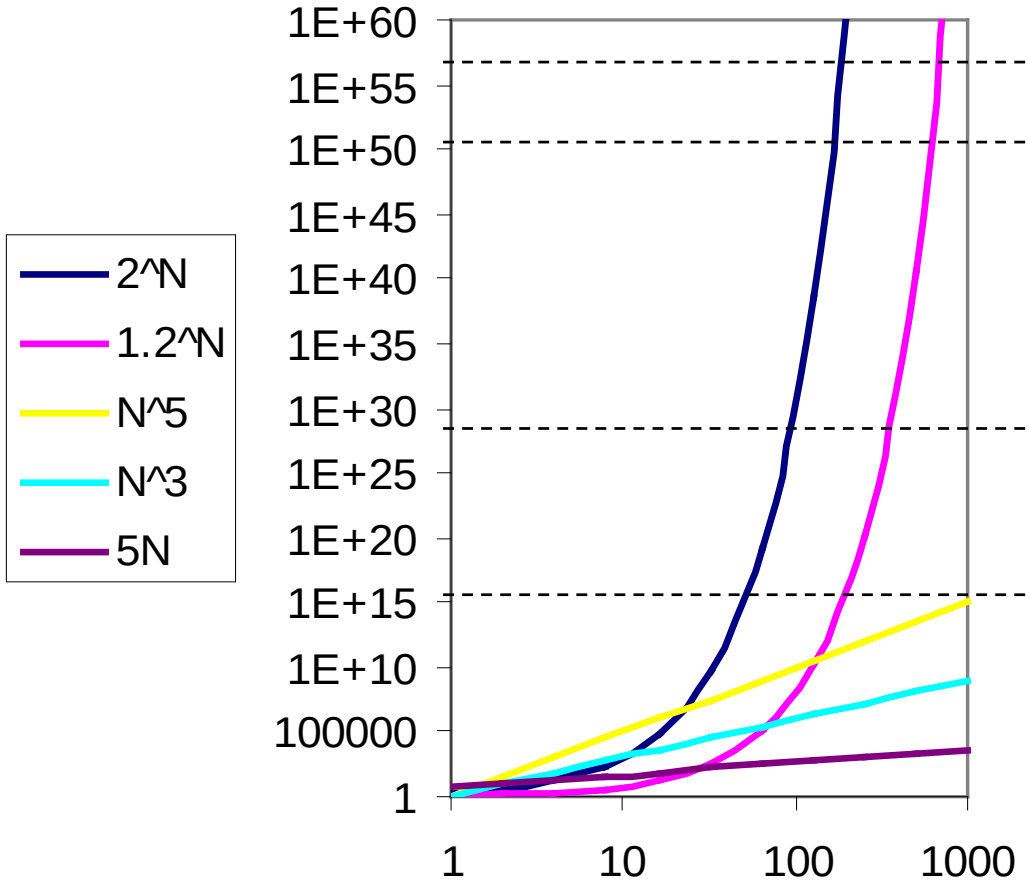
The Ultimate Laptop

Seth Lloyd, SCIENCE, 31 Aug 2000

"Using a black hole as a data storage and/or computing device"

5.4×10^{50} operations per second

What a Better Laptop Buys You



1 year

1 second

Ultimate Laptop

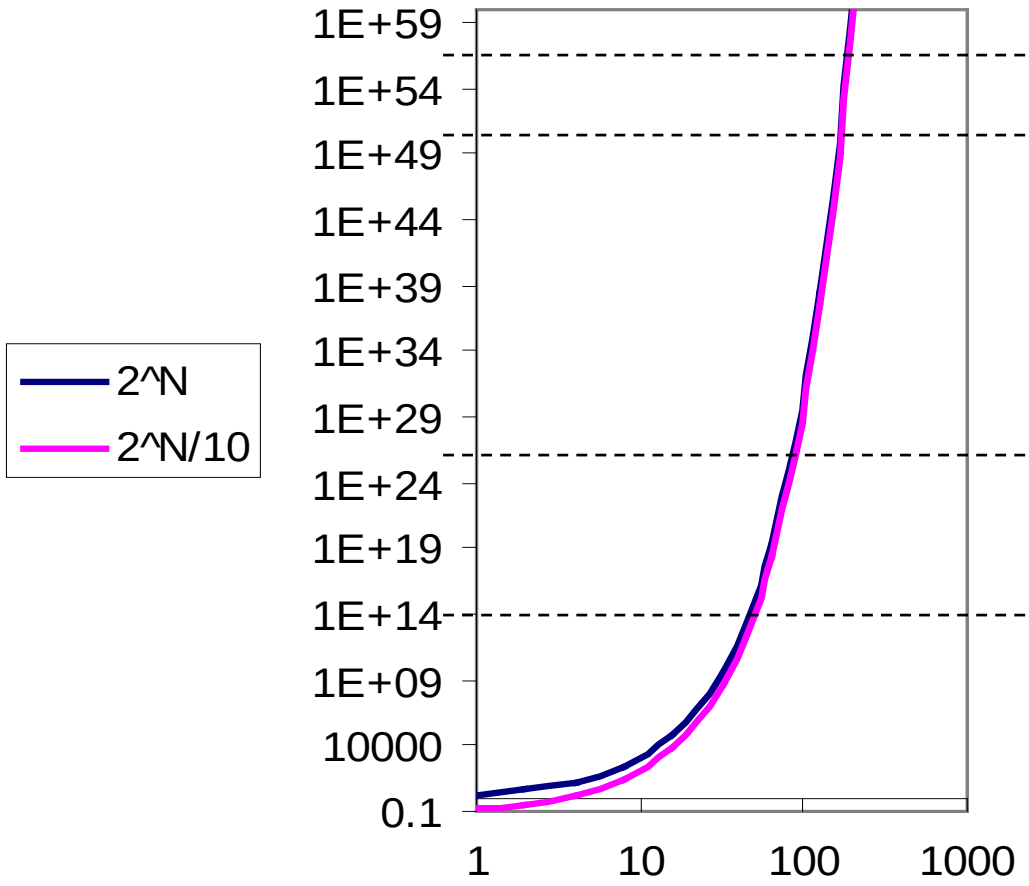
Since Big Bang

1 day

Today's laptop (100kMIPS)

Not much...

What Better Coding Buys You



1 year

1 second

Ultimate Laptop,

Since Big Bang

1 day

Today's laptop (1000MIPS)

Ten times faster buys you…

…nothing

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} years
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years	very long
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years	very long
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

Specific Goals of the Course

- Become familiar with some of the fundamental data structures in computer science
- Improve ability to solve problems abstractly
  - data structures are the building blocks
- Improve ability to analyze your algorithms
  - prove correctness
  - gauge (and improve) time complexity

Abstract Data Types

Abstract Data Type (ADT)

Mathematical description of an object and the set of operations on the object, e.g., list

Implement

Concrete Data Type

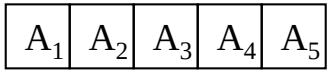
Specific Data Types or Data Structures, e.g., integer, array, linked list, …

Example: List ADT

- Mathematical description: a *sequence* of items
  - A_1 , A_2 , A_3 , A_4 , … , A_n
  - A_i precedes A_{i+1} for $1 \leq i < n$
- Operations
  - Length() = integer
  - Value(position) = item
  - Insert(position)
  - Delete(position)
  - Set(position, item)

Example: List ADT

- Array implementation



- Operations
  - Length() O(1)
  - Value(position) O(1)
  - Insert(position) O(n)
  - Delete(position) O(n)
  - Set(position, item) O(1)