Introduction to Computation

ESR Programming Workshop - IGNITE

Killian Smith

Winter 2018-19

Introduction

- This is meant as a quick and interactive overview of data structures and algorithms.
- If you see something you are interested in we can spend more time on it.
- Feel free to interrupt if you have questions or comments. We have lots of time.

Algorithms

The step-by-step procedure of solving a problem. Roughly speaking, they follow one of the following patterns:

- Divide and Conquer (inductive)
- Greedy (optimal subproblems)
- Dynamic (recurrence relations)
- Brute Force / Heuristic / Statistical

Measuring Performance

- Clock time is not a good measure. Why?

Measuring Performance

- Clock time is not a good measure. Why?
  - Machine hardware, architecture, running software, network connection, etc, are all variables

Measuring Performance

- Clock time is not a good measure. Why?
  - Machine hardware, architecture, running software, network connection, etc, are all variables
- Asymptotic Complexity
  - Limit as size of input n approaches ∞
  - Time or Space

Asymptotic Complexity

We care about the number of steps taken given an arbitrary input(s).

- $O(f) \rightarrow$ function f is an *upper* bound
- $\Theta(f) \rightarrow$ function f is an *exact* bound
- $\Omega(f) \rightarrow$ function f is a *lower* bound

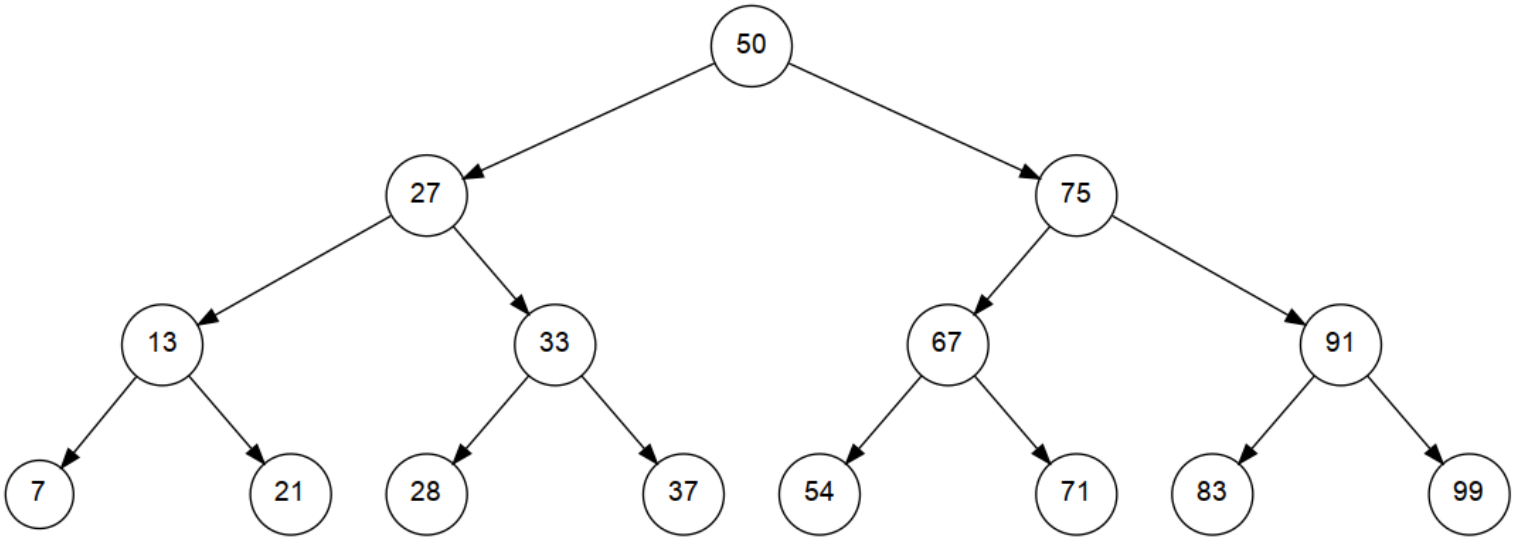
List Example

Given a list of n elements in $\mathbb{Z}$, how many steps are needed to compute the sum?

- Initialize our algorithm with a sum of 0 as an accumulator.
- Add the x_0 element to the accumulator
  - Then add the x_1 element to accumulator
  - Then add the x_2 element to the accumulator
  - ...
- This repeats **exactly** n times: therefore this algorithm is $\Theta(n)$.
  - We can also call this a *linear* algorithm.

BST Example

Given a *Binary Search Tree* of n elements in $\mathbb{Z}$, how many steps are needed to search for a given $x \in \mathbb{Z}$? Assume that the tree is perfectly balanced.



BST Example (cont)

- Start at the root node that contains element k.
  - If $x = k$, return *true*.
  - If $x < k$, go to the left child.
  - If $x > k$, go to the right child.
- Repeat until x is found or *empty* tree is reached.

BST Example (cont)

How many steps is this compared to the tree size of n?

Consider when the worst case occurs.

BST Example (cont)

How many steps is this compared to the tree size of n?

Consider when the worst case occurs.

- When we travel the whole *height* of the tree.
  - A tree of height 1 can have 1 element
  - A tree of height 2 can have 3 elements
  - A tree of height 3 can have 7 elements

Which means...

BST Example (cont)

How many steps is this compared to the tree size of n?

Consider when the worst case occurs.

- When we travel the whole *height* of the tree.
  - A tree of height 1 can have 1 element
  - A tree of height 2 can have 3 elements
  - A tree of height 3 can have 7 elements

Which means...

- A tree of height h can have $2^h - 1$ elements

BST Example (cont)

- Rewrite in terms of n.

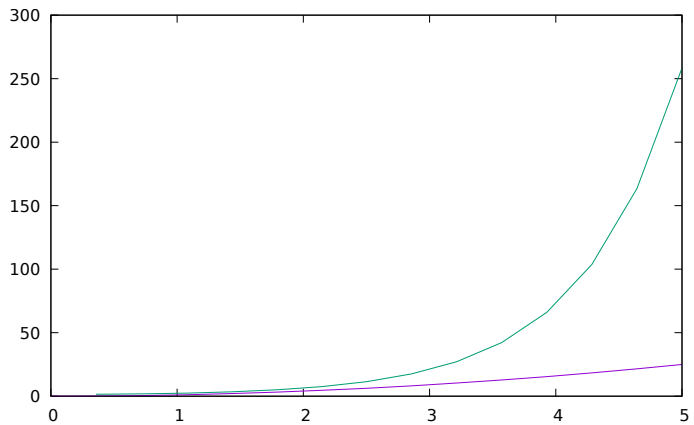
$$
\begin{aligned}
2^h - 1 &= n \\
2^h &= n + 1 \\
h &= log_2(n+1) \\
h &= log(n) \qquad \leftarrow \text{what happened here?}
\end{aligned}
$$

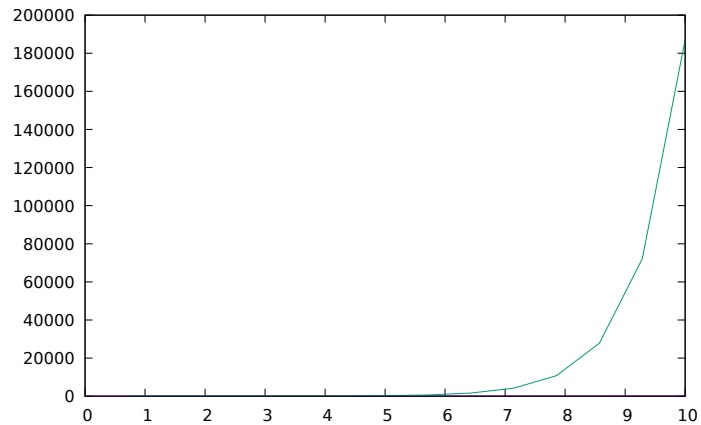
- This is the *worst case* for a *balanced* tree, so we would say the complexity is $O(log(n))$, or has a logarithmic running time.
- What happens if the tree is not balanced?

Importance of Complexity Analysis

String Alignment

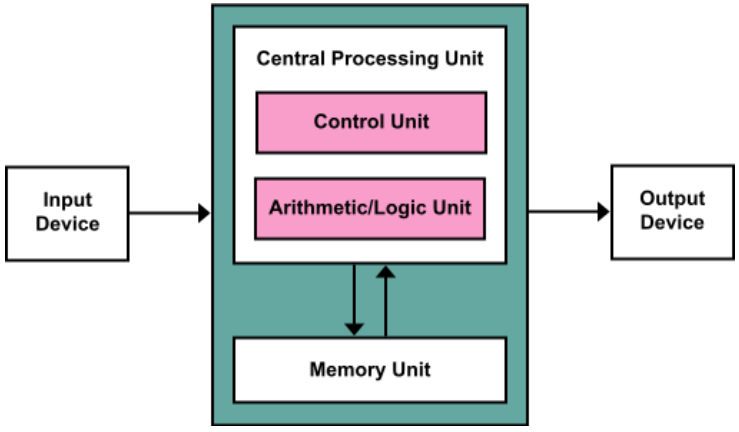
- Brute Force $\in NP := \binom{m+n}{n} = O(\frac{(m+n)!}{(n!)^2})$
- Dynamic (Needleman-Wunsh) $\in P := O(mn)$





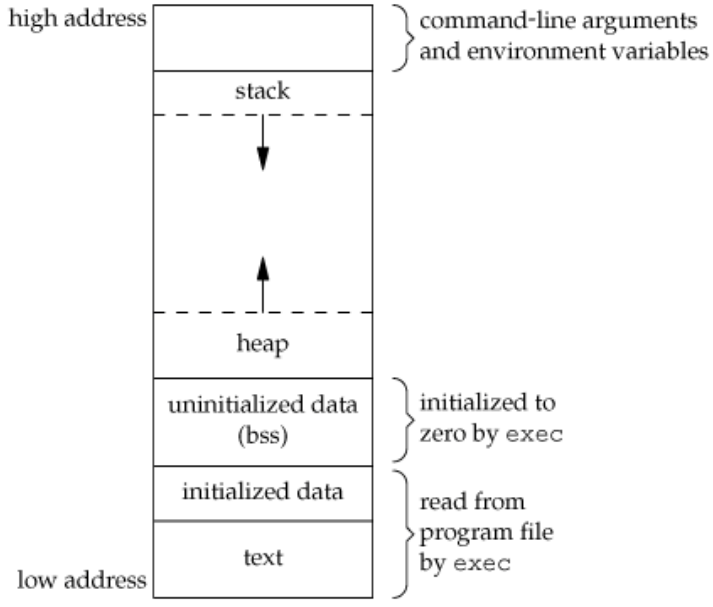
Machine

Hardware



Source : Kapooht - CC BY-SA 3.0

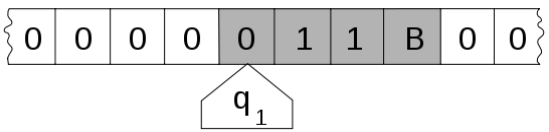
Memory



Source : Holberton School

What is Computable?

Turing Machine



Source : wikipedia.org

Lambda Calculus

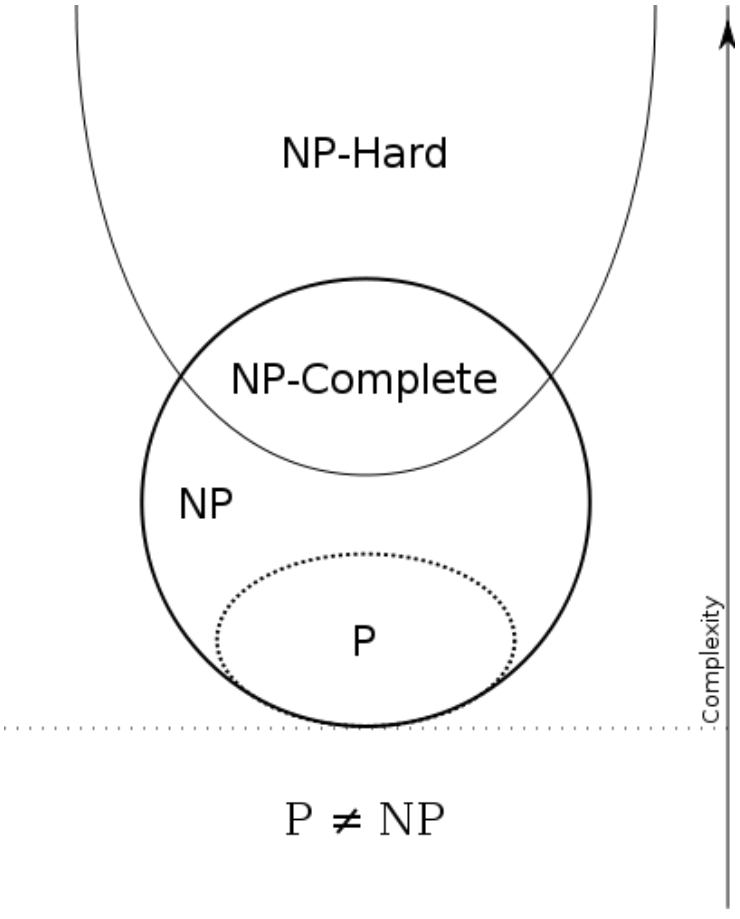
$$0 := \lambda f.\lambda x.x$$

$$1 := \lambda f.\lambda x.fx$$

$$2 := \lambda f.\lambda x.f(fx)$$

$$3 := \lambda f.\lambda x.f(f(fx))$$

What is Computable? (cont)

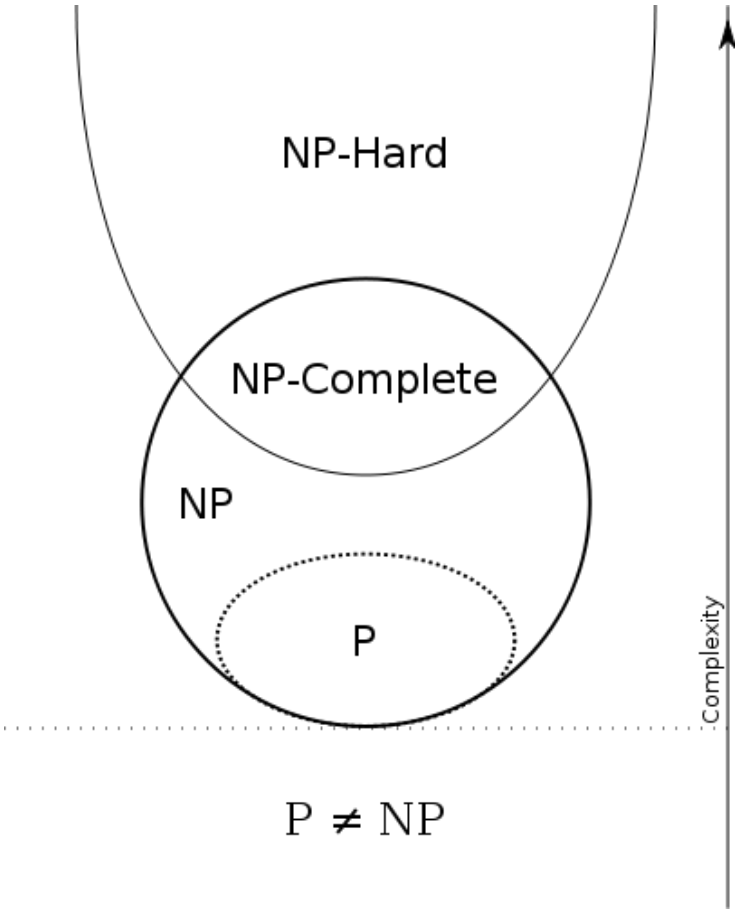


Source : wikipedia.org

Problems in P

- Polynomial time *computable*
- Solvable by a deterministic Turing machine or primitive recursion
- Max element, sorting....

What is Computable? (cont)

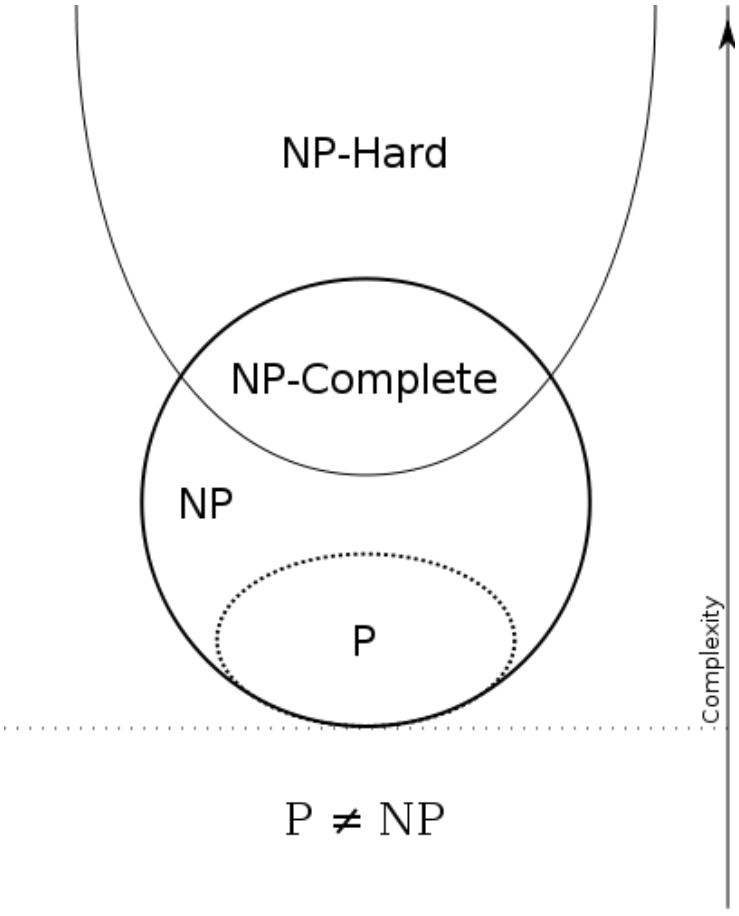


Source : wikipedia.org

Problems in *NP*

- Polynomial time *verifiable*
- Solvable by a non-deterministic Turing machine or general recursion
- 3-SAT, traveling salesman...

What is Computable? (cont)



Source : wikipedia.org

Problems in *NP-Hard*

- At least as hard as NP-Complete.
- Halting Problem, Max Clique...

Computability in Bioinformatics

Problems $\in P$

- Relative Amino Abundance
- String Alignment
- String Search

Problems $\notin P$

- Sequence Assembly
- Breakpoint Distance
- Topology of Pangenomes

Data Types / Data Structures

Computer memory is *linear* and cells are *fixed* size. Real world data needs to be parsed into a format that a machine can operate over. Data can be constructed with:

- Unboxed raw values (machine words)
- Boxed values with a type (pointer to data)
- Product of datum (**and**)
- Coproduct of datum (**or**)

Data Types / Data Structures (cont)

There are 2 special types:

- Empty Type (0) - No inhabitants
- Unit Type (1) - Single inhabitant

We can also leave "holes" in data to fill in later. This is called *polymorphism*.

Linked List

We define a list inductively: it can either be empty or some element and the rest of the list. Mathematically, we could write

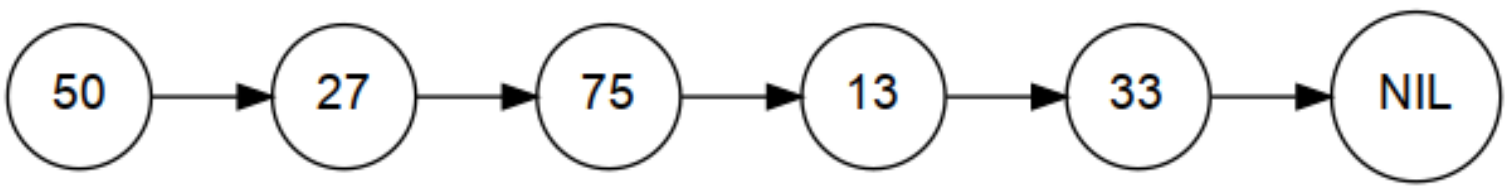
$$L_a = 1 + a * L_a$$

where a is the placeholder for the type of data that will eventually be in the list (i.e.- polymorphism).

Linked List (cont)

To decompose to a machine representation, determine what the building blocks of the structure are. In this case:

- We can make a unit type (the *Nil* or empty list).
- Or we can make a pair of a value $x \in a$ and another list structure (which is a pointer)



Control

Data Types mean nothing without proper control structures for manipulation. Machines have many variations of *JMP*. This is bad for actual software!

Control (cont)

Imperative remedy this with:

- Conditional Branching (if-then-else, switch)
- Semi-Structured Looping (while, do-while, for-loop)

Declarative remedy this with:

- Conditional Branching (if-then-else)
- Pattern Matching
- Recursion

Exercises (1)

Max Element in Linked List

- Algorithm
- Complexity

Exercises (2)

Max Element in a BST

- Algorithm
- Complexity

Exercises (3)

Sorting a List

- Bubble/Insertion Sort
- Merge Sort
- Compare

Exercises (3)

Fractional Knapsack Problem

Given a bag that can hold W kilograms, and piles of spices $S_1, S_2 ... S_n$, which have a price value of $V_1, V_2 ... V_n$, find an algorithm to maximize the value of the spices contained in the bag.

- Algorithm
- Complexity

End

Questions or Comments?