Welcome to CS106B!

Who's Here Today?

- Aero/Astro
- Anthropology
- Art Practice
- Bioengineering
- Biology
- Business
- Chemical Engineering
- Chemistry
- Civil/Environmental Engineering
- Creative Writing
- Data Science
- East Asian Studies
- Economics
- Education
- Electrical Engineering
- Energy Resources Engineering
- English
- Environmental Systems Engineering
- FemGen
- Genetics
- History
- Human Biology
- Immunology
- International Relations
- Law
- Materials Science
- Mechanical Engineering
- Microbiology and Immunology
- Middle Eastern Languages / Culture
- MS&E
- Physics
- Political Science
- Product Design
- Psychology
- Public Policy
- Spanish
- Statistics
- STS
- Symbolic Systems
- SymSys
- ***Undeclared!***

Course Staff

Instructor: Keith Schwarz
(htiek@cs.stanford.edu)

Head TA: Kate Rydberg
(rydbergk@stanford.edu)

The CS106B Section Leaders
The CS106B Course Helpers

Course Website

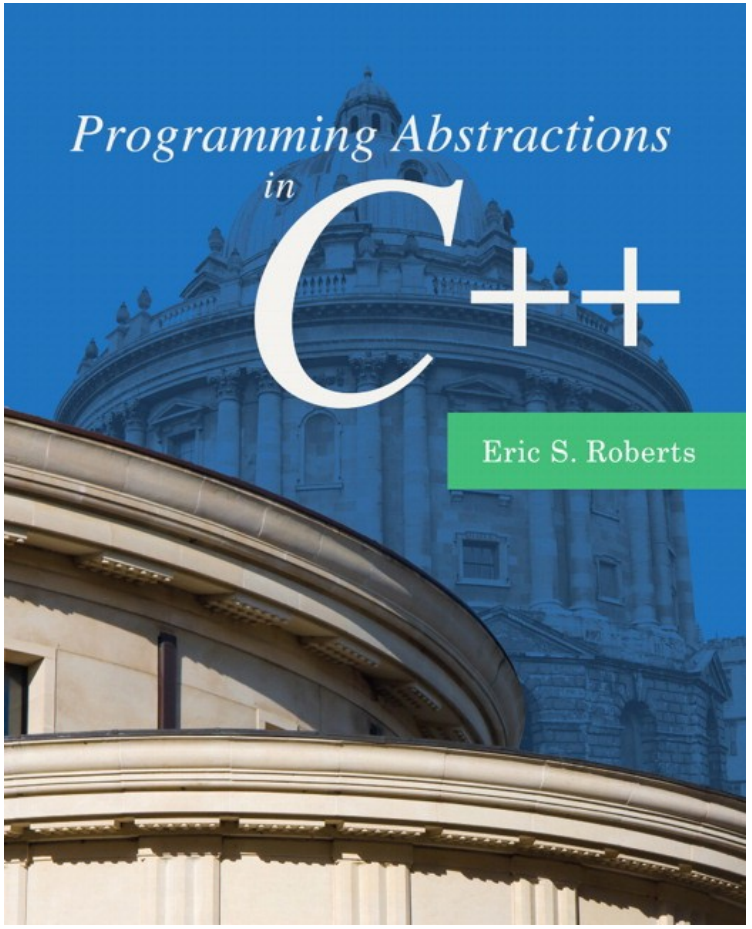
https://cs106b.stanford.edu

Prerequisites

CS106A

(or equivalent)
(check out our course placement handout if you're unsure!)

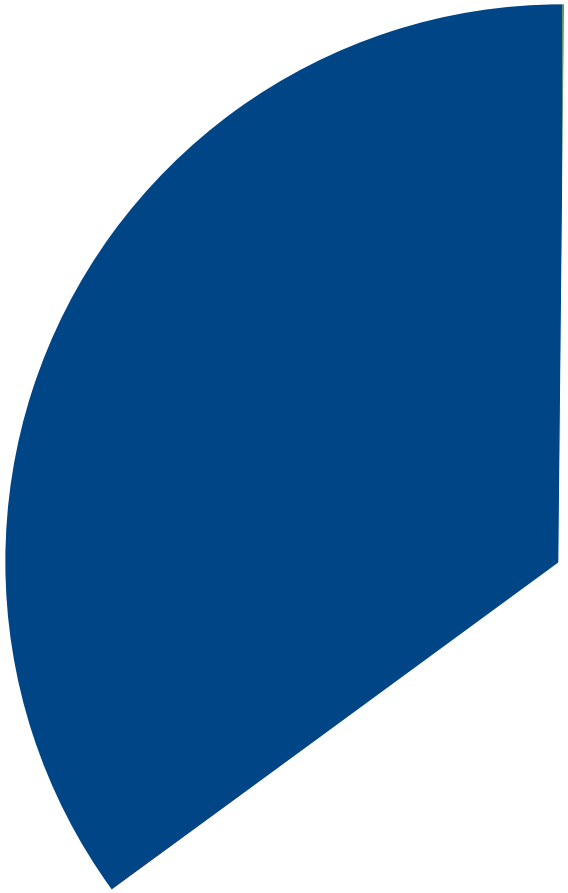
Required Reading



- Available in the bookstore. Some copies are on reserve in the Engineering library.
- We do recommend picking up a copy of this book, since it provides a lot of useful extra background information.

Grading Policies

Grading Policies



35% Assignments

Eight Assignments

(One intro assignment that goes out today, seven programming assignments)

Grading Policies

35% Assignments

25% Midterm Exam

Midterm Exam

Tuesday, February 19th
7PM – 10PM
Location TBA

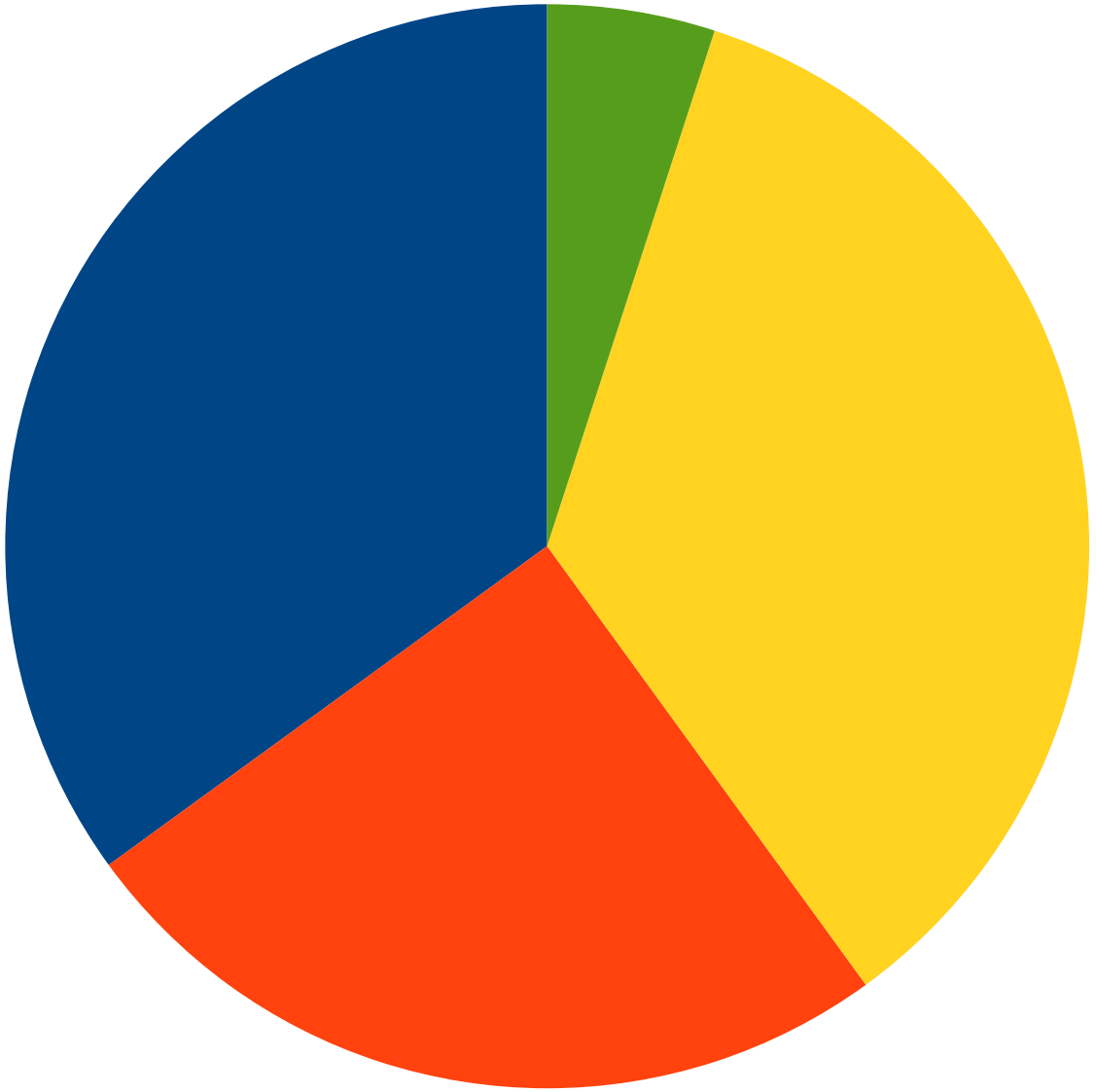
Grading Policies

- 35% Assignments
- 25% Midterm Exam
- 35% Final Exam

Final Exam

Monday, March 18th
8:30AM – 11:30AM
No alternate exams except for OAE accommodations.

Grading Policies



- 35% Assignments
- 25% Midterm Exam
- 35% Final Exam
- 5% Section Participation

Discussion Sections

Weekly sections. Let’s go talk about them!

Discussion Sections

- There are weekly discussion sections in CS106B. Section attendance is required.
- Sign up between Thursday, January 10th at 5:00PM and Sunday, January 13th at 5:00PM by visiting

  http://cs198.stanford.edu/section

- We don't look at Axess for section enrollments. Please make sure to sign up here even if you're already enrolled on Axess.

CS106S

- CS106S is an optional one-unit add-on course for CS106B that touches on applications of the material to civics, education, healthcare, and the like.
- This is “in addition to” rather than “instead of” regular section.e

How Many Units?

```
int numUnits(bool isGrad) {
    if (isGrad) {
        return randomInteger(3, 5); // 3 to 5
    } else {
        return 5;
    }
}
```

Getting Help

Getting Help

- LaIR Hours!
  - Sunday – Thursday, 7PM – 11PM
  - Held in the first floor of Tresidder Student Union.
  - LaIR hours start next week.
- Kate’s Office Hours in Gates B02
  - Tuesdays and Thursdays, 1:30PM – 2:30PM.
- Keith's Office Hours in Gates 178
  - Tuesdays, 10:00AM – 12:00PM.
  - Stop on by! I’m happy to chat about just about anything.

What's Next in Computer Science?

Goals for this Course

- ***Learn how to model and solve complex problems with computers.***
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Quantitatively analyze different approaches for solving problems.

Goals for this Course

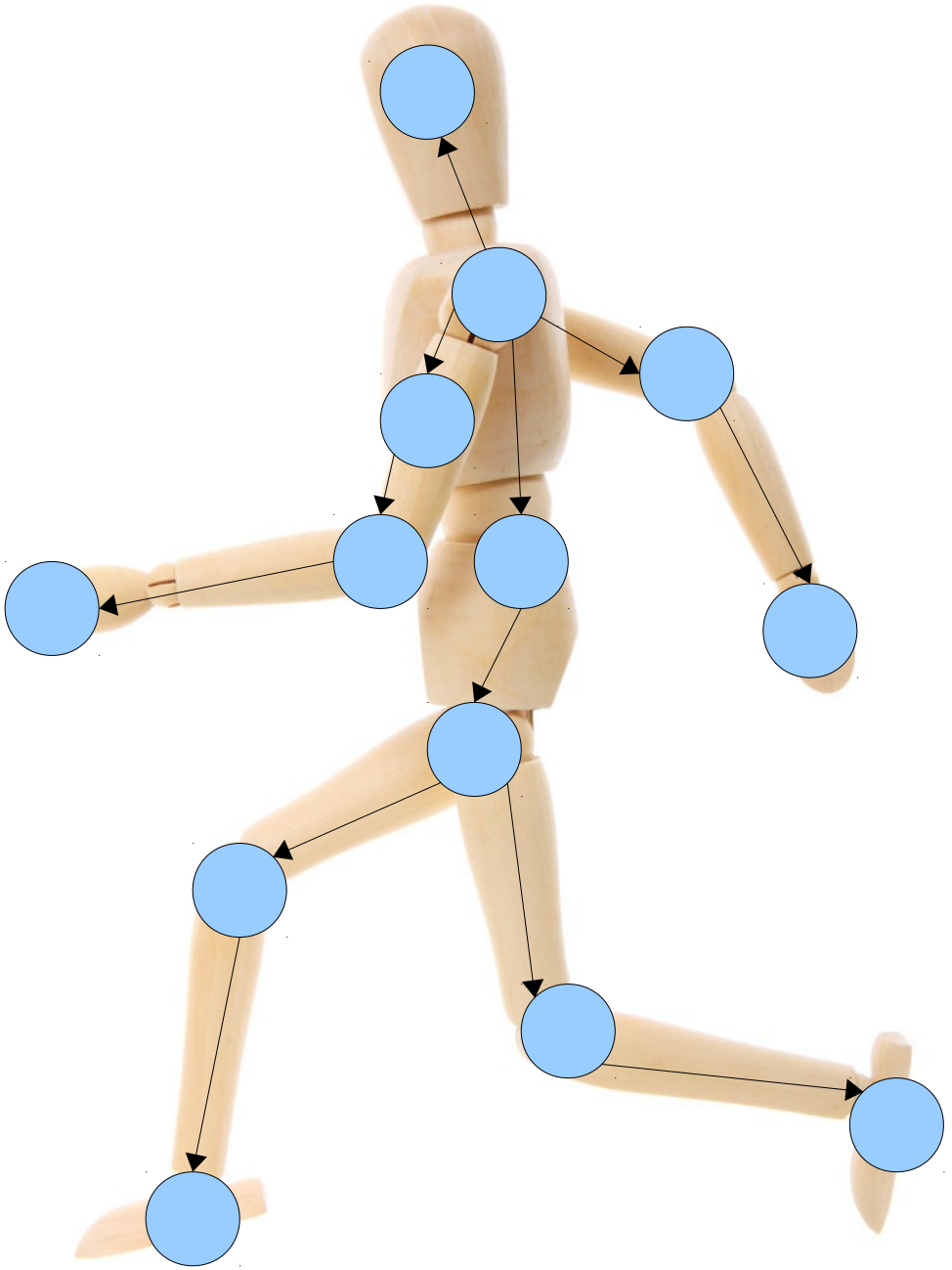
Learn how to model and solve complex problems with computers.

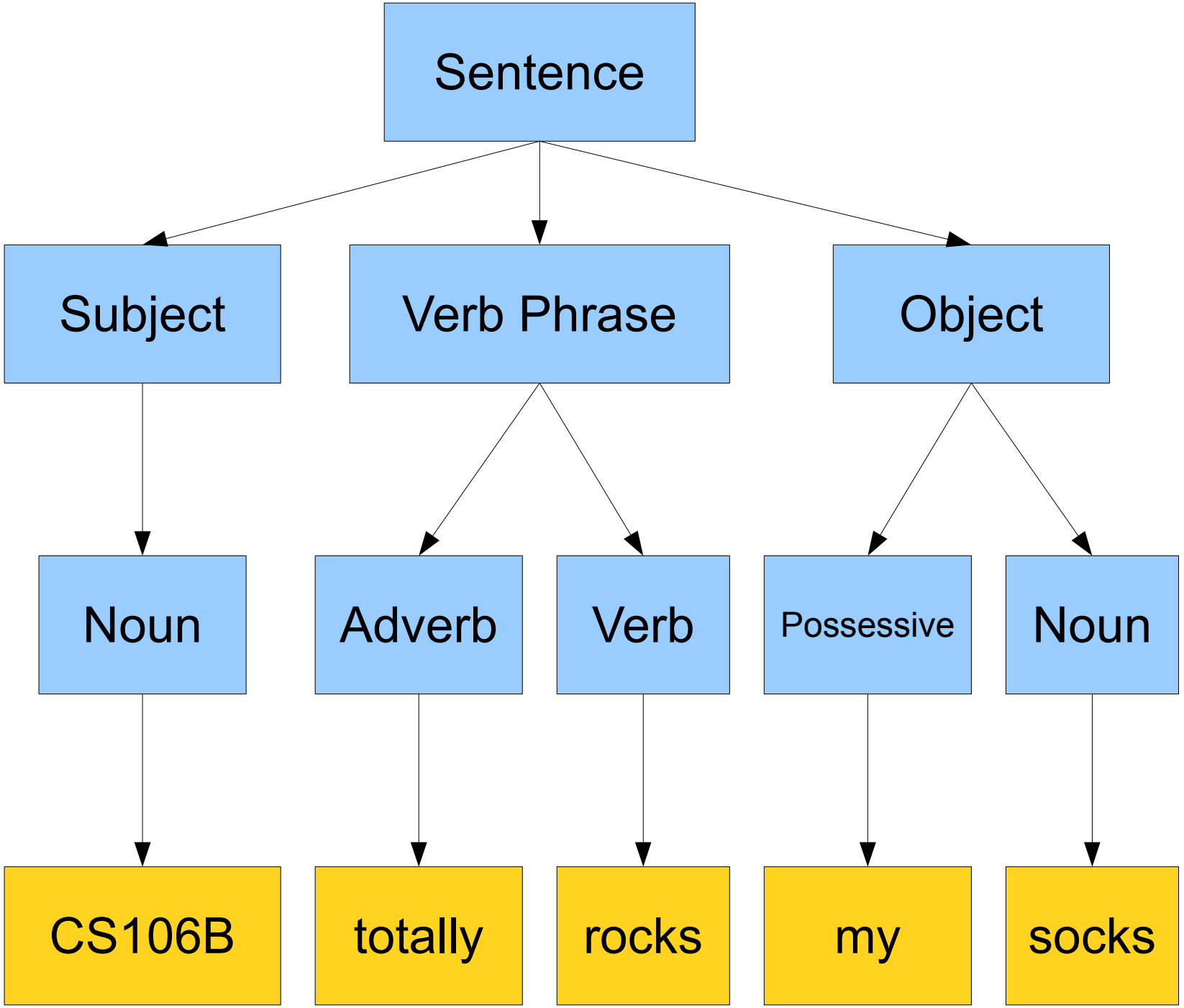
To that end:

- Explore common abstractions for representing problems.

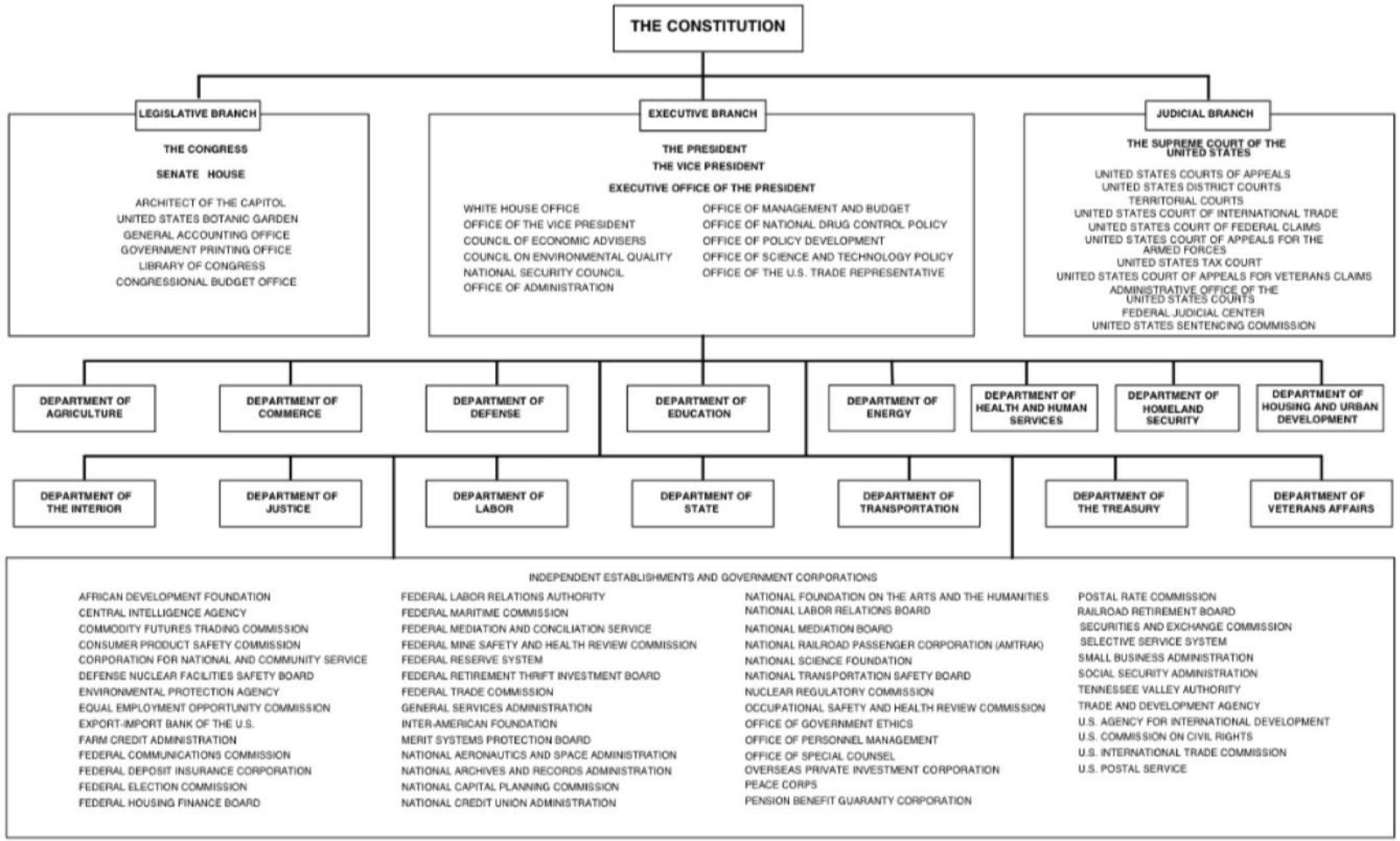
  Harness recursion and understand how to think about problems recursively.

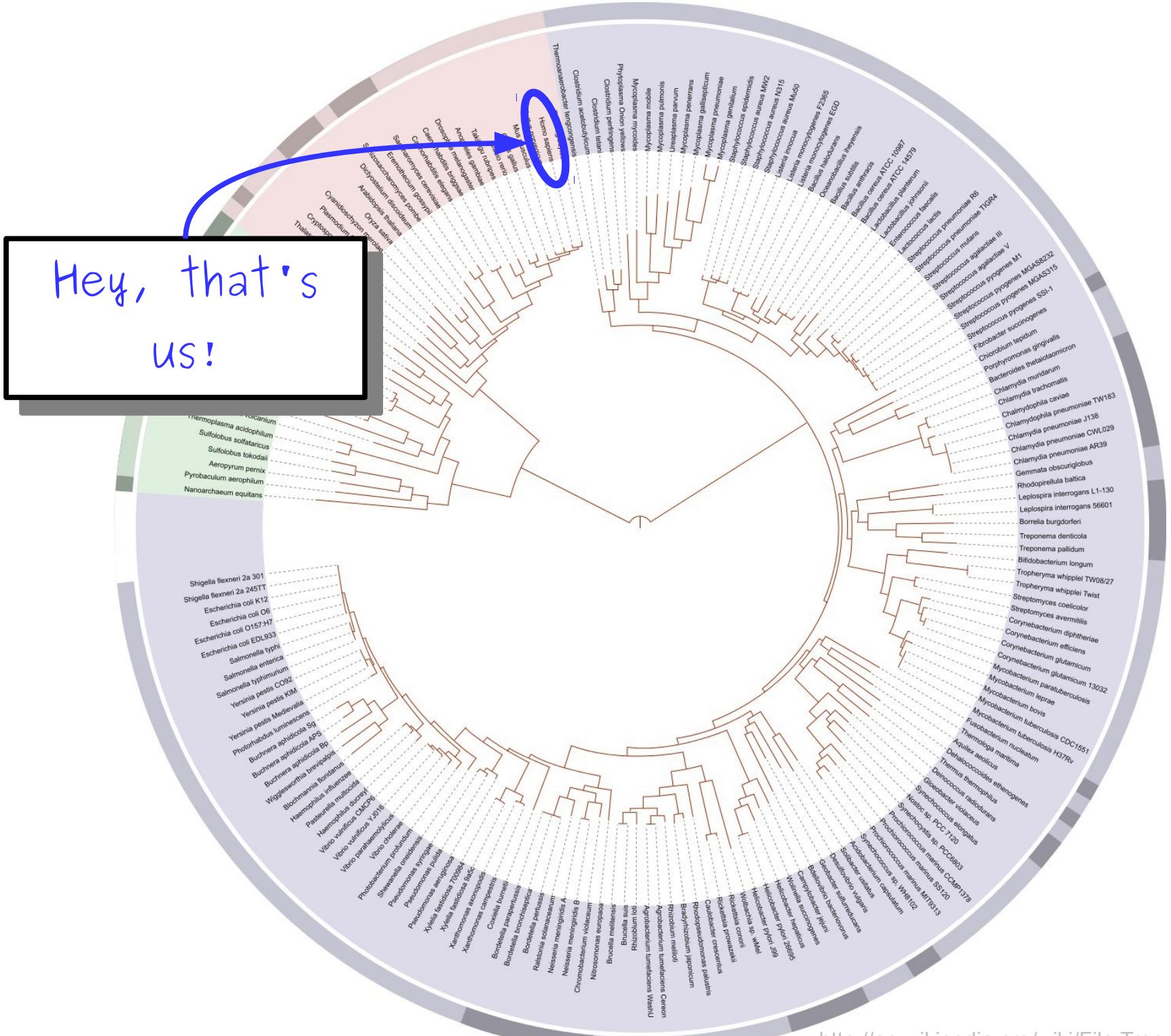
  Quantitatively analyze different approaches for solving problems.

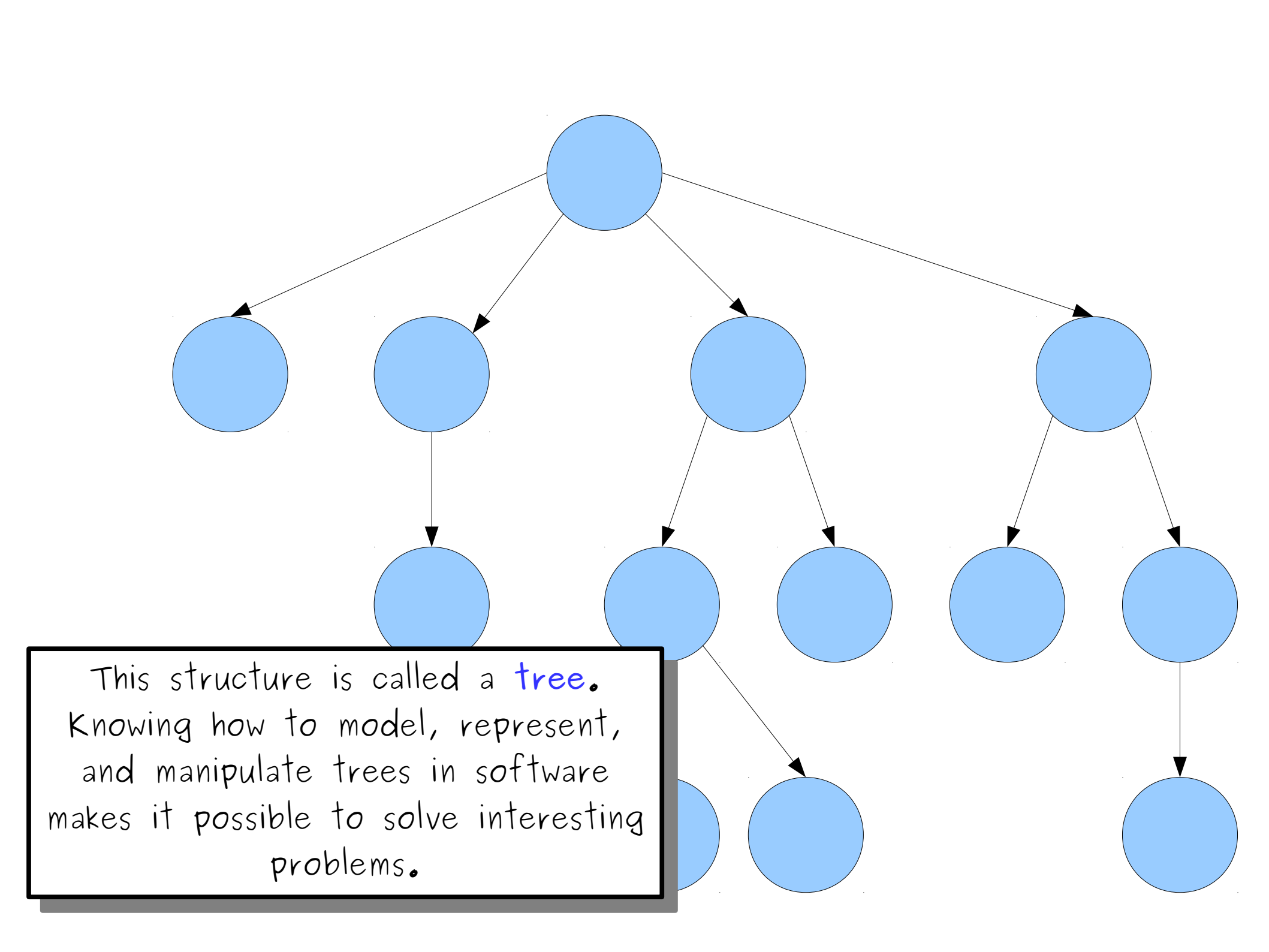



Sentence
Subject
Verb Phrase
Object
Noun
Adverb
Verb
Possessive
Noun
CS106B
totally
rocks
my
socks

THE GOVERNMENT OF THE UNITED STATES

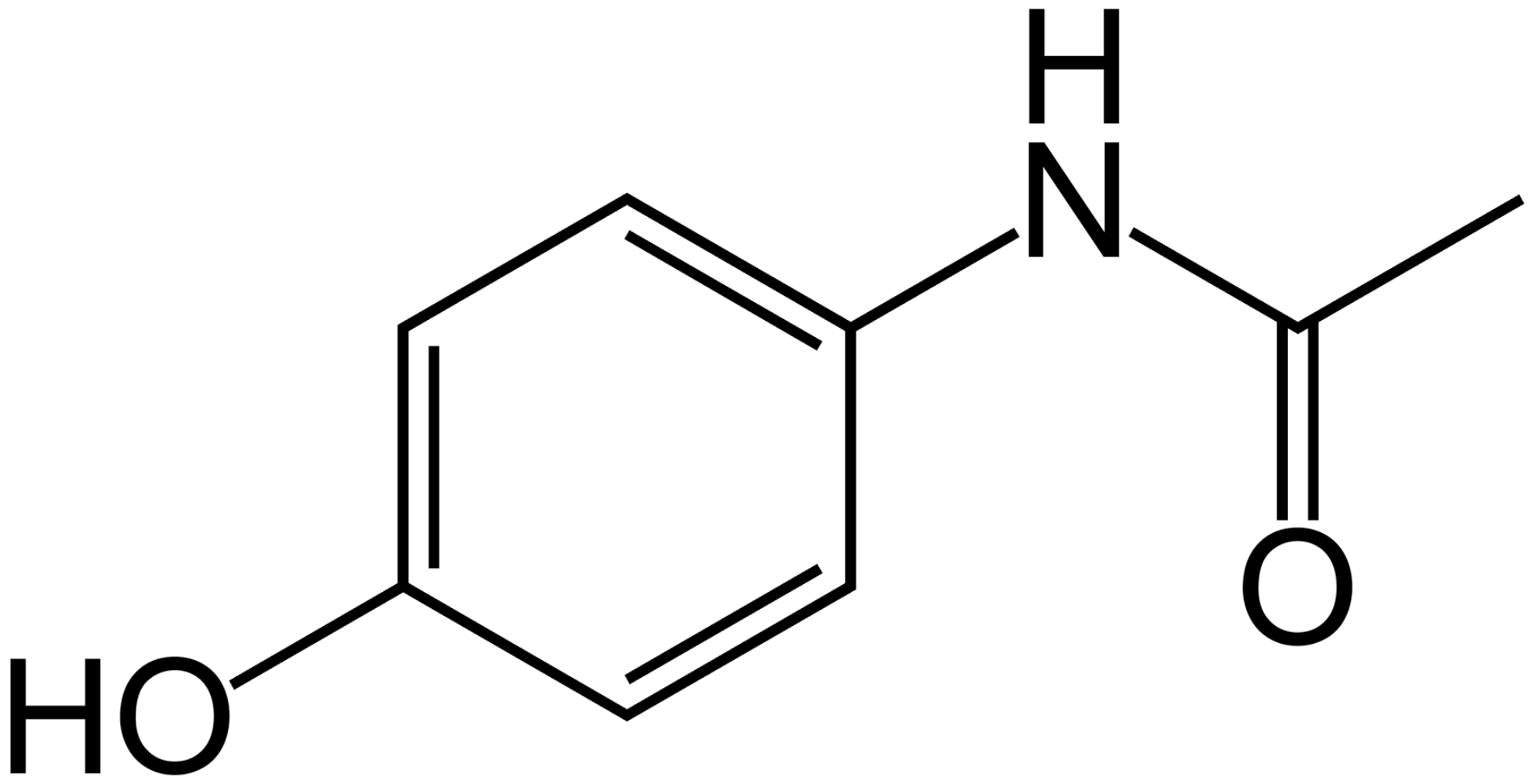




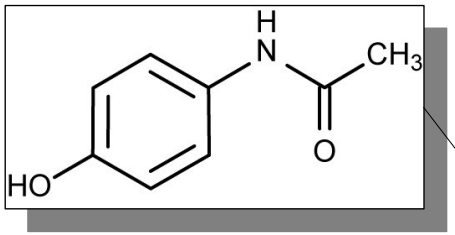

This structure is called a tree.
Knowing how to model, represent,
and manipulate trees in software
makes it possible to solve interesting
problems.

Building a vocabulary of ***abstractions*** makes it possible to represent and solve a wider class of problems.

How do we keep passwords secure when servers are hacked all the time?



How do we quickly check whether a chemical has already been discovered?



Hash Function

553872289012

224224651111

Inputs can be just about anything: strings, ID numbers, molecular shapes, passwords, etc.

Output is a seemingly random number that serves as a "fingerprint" of the input.

Building a vocabulary of ***abstractions*** makes it possible to represent and solve a wider class of problems.

Goals for this Course

- ***Learn how to model and solve complex problems with computers.***
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Quantitatively analyze different approaches for solving problems.

Goals for this Course

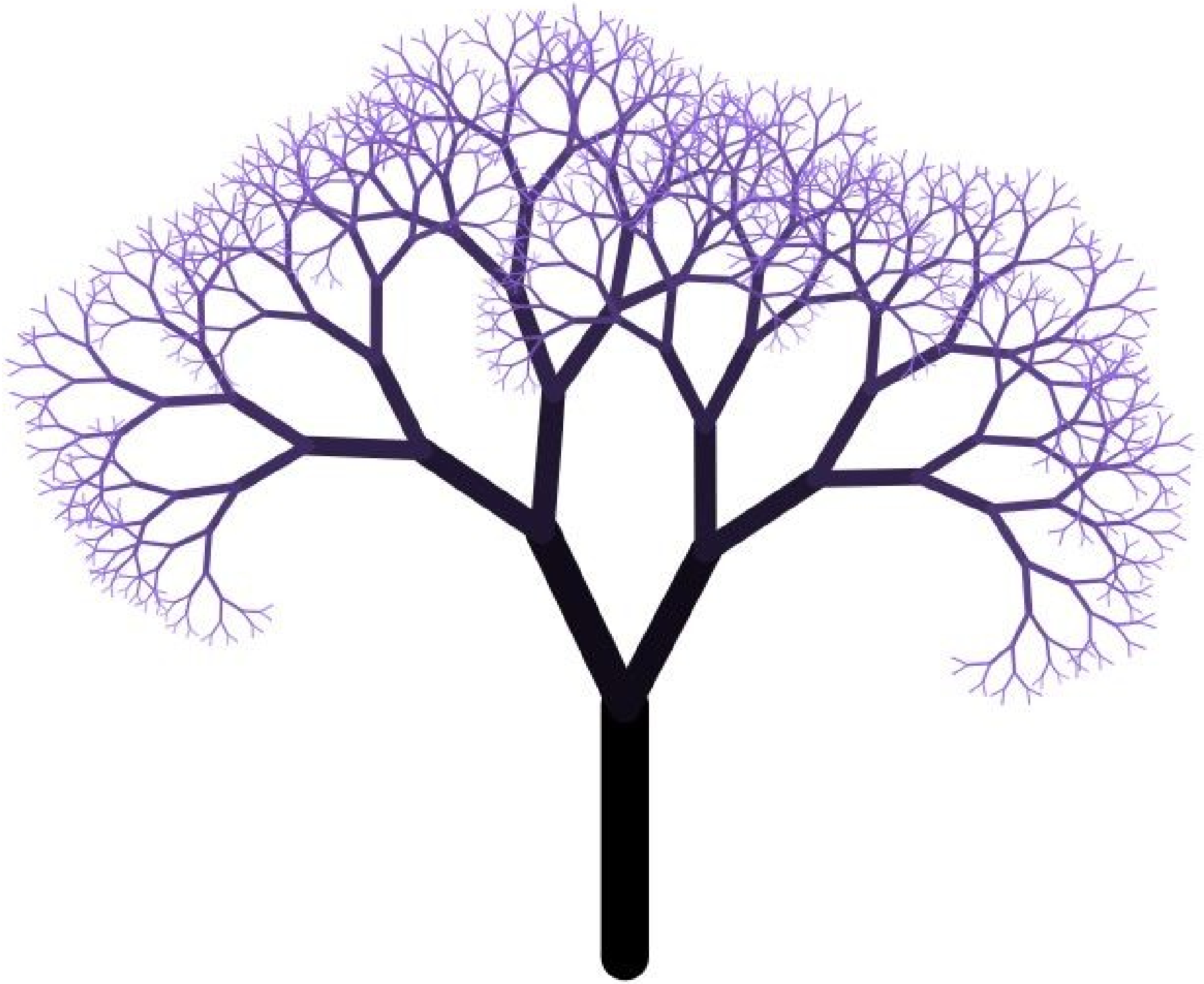
Learn how to model and solve complex problems with computers.

To that end:

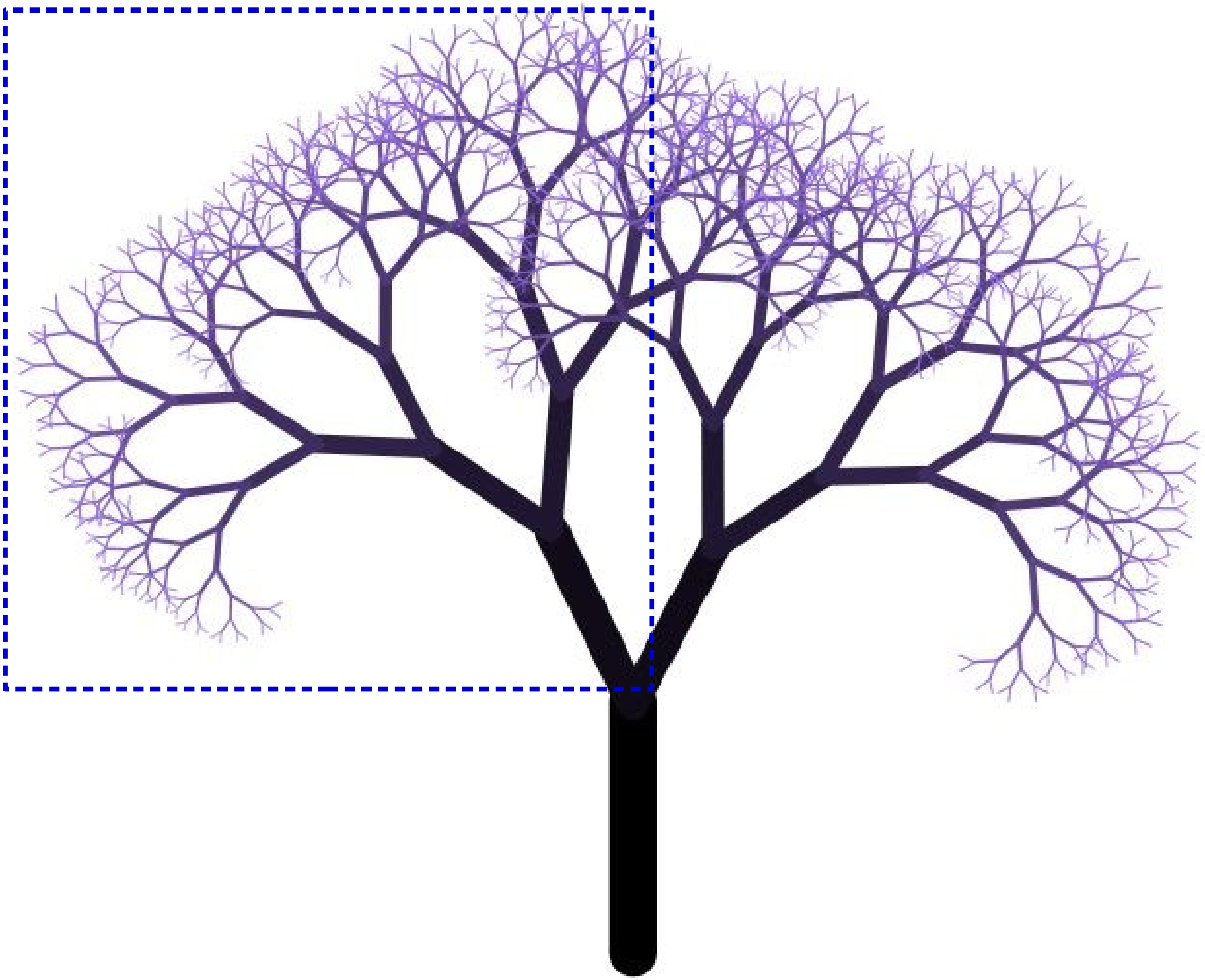
Explore common abstractions for representing problems.

- Harness recursion and understand how to think about problems recursively.

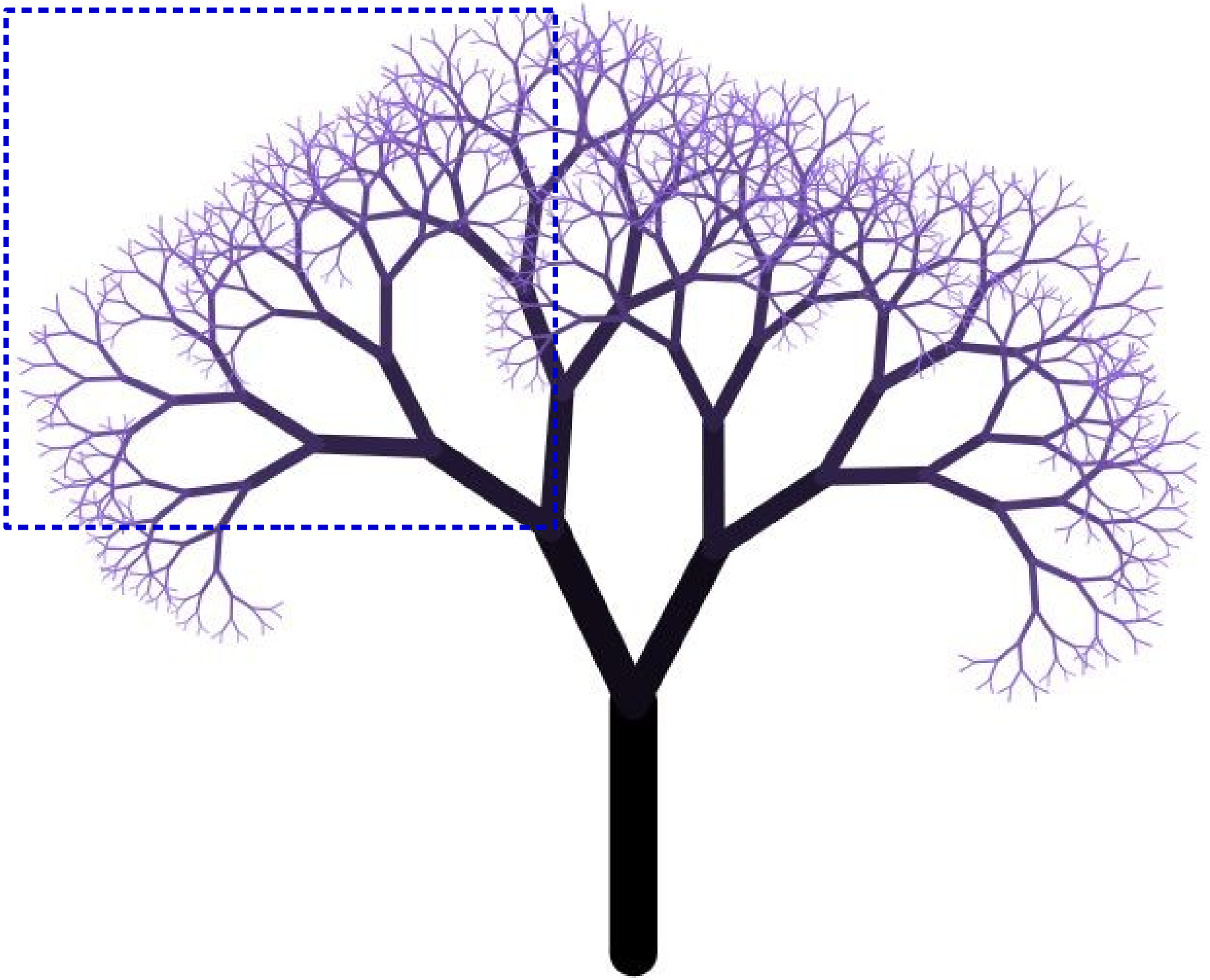
Quantitatively analyze different approaches for solving problems.



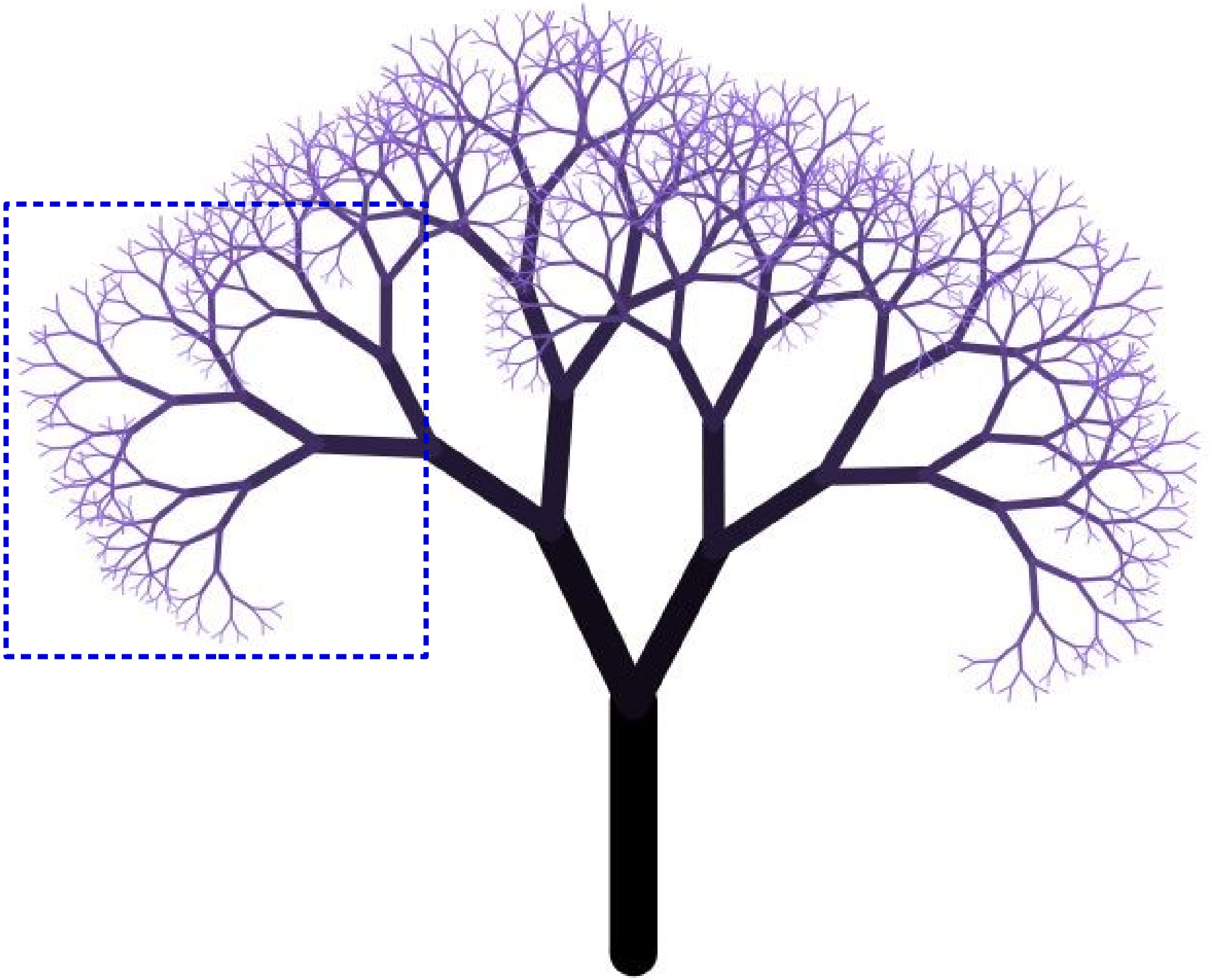
http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg



http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg



http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg



http://www.marketoracle.co.uk/images/2010/Oct/fractal-tree2.jpg

Creating Trees

A ***recursive solution*** is a solution that is defined in terms of itself.

Goals for this Course

- ***Learn how to model and solve complex problems with computers.***
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Quantitatively analyze different approaches for solving problems.

Goals for this Course

Learn how to model and solve complex problems with computers.

To that end:

Explore common abstractions for representing problems.

Harness recursion and understand how to think about problems recursively.

- Quantitatively analyze different approaches for solving problems.

ull,"status":"reviewed","tsunami":0,"sig":369,"net":"us","code":"2000j048","ids":",us2000j
origin,phase-data,","nst":null,"dmin":1.598,"rms":0.78,"gap":104,"magType":"mww","type":"e
Tobelo, Indonesia"},"geometry":{"type":"Point","coordinates":[127.3157,2.3801,53.72]},"id"
{"type":"Feature","properties":{"mag":5.1,"place":"265km SW of Severo-Kuril'sk,
Russia","time":1546548377590,"updated":1546549398040,"tz":600,"url":"https://earthquake.us
detail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us2000j03t.geojson","fel
,"status":"reviewed","tsunami":0,"sig":400,"net":"us","code":"2000j03t","ids":",us2000j03t
gin,phase-data,","nst":null,"dmin":5.198,"rms":0.94,"gap":48,"magType":"mww","type":"earth
Severo-Kuril'sk, Russia"},"geometry":{"type":"Point","coordinates":[153.7105,48.8712,104.7
{"type":"Feature","properties":{"mag":4.8,"place":"20km NNW of Taitung City,
Taiwan","time":1546538570070,"updated":1546541624040,"tz":480,"url":"https://earthquake.us
detail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us2000j02k.geojson","fel
,"status":"reviewed","tsunami":0,"sig":354,"net":"us","code":"2000j02k","ids":",us2000j02k
gin,phase-data,","nst":null,"dmin":0.52,"rms":0.79,"gap":110,"magType":"mb","type":"earthq
City, Taiwan"},"geometry":{"type":"Point","coordinates":[121.0489,22.9222,10]},"id":"us200
{"type":"Feature","properties":{"mag":5,"place":"79km ENE of Petropavlovsk-Kamchatskiy,
Russia","time":1546538266300,"updated":1546541474965,"tz":720,"url":"https://earthquake.us
detail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us2000j02g.geojson","fel
us":"reviewed","tsunami":0,"sig":385,"net":"us","code":"2000j02g","ids":",us2000j02g,","so
in,phase-data,","nst":null,"dmin":0.728,"rms":0.75,"gap":114,"magType":"mb","type":"earthq
Petropavlovsk-Kamchatskiy, Russia"},"geometry":{"type":"Point","coordinates":[159.6844,53.
{"type":"Feature","properties":{"mag":4.5,"place":"South of Java,
Indonesia","time":1546533739000,"updated":1546539809085,"tz":420,"url":"https://earthquake
","detail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us2000j024.geojson","
tatus":"reviewed","tsunami":0,"sig":312,"net":"us","code":"2000j024","ids":",us2000j024,",
rigin,phase-data,","nst":null,"dmin":2.821,"rms":0.89,"gap":83,"magType":"mb","type":"eart
Indonesia"},"geometry":{"type":"Point","coordinates":[108.5165,-10.6419,8.84]},"id":"us200
{"type":"Feature","properties":{"mag":4.8,"place":"108km N of Ishigaki,
Japan","time":1546529675300,"updated":1546530815040,"tz":480,"url":"https://earthquake.usg
etail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us2000j01x.geojson","felt
"status":"reviewed","tsunami":0,"sig":354,"net":"us","code":"2000j01x","ids":",us2000j01x,
in,phase-data,","nst":null,"dmin":1.342,"rms":0.82,"gap":68,"magType":"mb","type":"earthqu
Japan"},"geometry":{"type":"Point","coordinates":[124.1559,25.3209,122.33]},"id":"us2000j0
{"type":"Feature","properties":{"mag":5.4,"place":"82km S of Bristol Island, South Sandwic
Islands","time":1546519662810,"updated":1546520523040,"tz":-120,"url":"https://earthquake.
,"detail":"https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us2000j01b.geojson","f

$$\frac{\frac{a}{b}}{\frac{c}{d}}$$

Source: https://datacenterfrontier.com/year-hyperscale-facebook-growth-innovation/

Goals for this Course

- ***Learn how to model and solve complex problems with computers.***
- To that end:
  - Explore common abstractions for representing problems.
  - Harness recursion and understand how to think about problems recursively.
  - Quantitatively analyze different approaches for solving problems.

Who's Here Today?

- Aero/Astro
- Anthropology
- Art Practice
- Bioengineering
- Biology
- Business
- Chemical Engineering
- Chemistry
- Civil/Environmental Engineering
- Creative Writing
- Data Science
- East Asian Studies
- Economics
- Education
- Electrical Engineering
- Energy Resources Engineering
- English
- Environmental Systems Engineering
- FemGen
- Genetics
- History
- Human Biology
- Immunology
- International Relations
- Law
- Materials Science
- Mechanical Engineering
- Microbiology and Immunology
- Middle Eastern Languages / Culture
- MS&E
- Physics
- Political Science
- Product Design
- Psychology
- Public Policy
- Spanish
- Statistics
- STS
- Symbolic Systems
- SymSys
- ***Undeclared!***

Speaking to the Computer

C++

What is C++?

- C++ is a programming language used to design complex, high-performance systems.
- C++ is an influential language. Java inherited much of its syntax from C++, and JavaScript retains many of its traits.
- There are many features of C++ that aren't present in Java / JavaScript / Python, and those features make it an attractive language for use in CS106B.
- C++ is a ***huge*** language that's undergone many revisions (it was invented in 1983; most recent version is C++17) and we won't be covering it in full depth. Take CS106L or CS110 for more!

```
/* File: hello-world.cpp
 *
 * A canonical Hello, world! program
 * in C++.
 */

#include <iostream>
using namespace std;

int main() {
    cout << "Hello, world!" << endl;
    return 0;
}
```

```
/* File: retain-evens.cpp
 *
 * A program to filter out odd numbers from a list.
 */
#include <iostream>
#include "vector.h"
using namespace std;

Vector<int> evensIn(Vector<int> values) {
    Vector<int> result;
    for (int i = 0; i < values.size(); i++) {
        if (values[i] % 2 == 0)
            result += values[i];
    }
    return result;
}

int main() {
    Vector<int> values = { 1, 2, 3, 4, 5 };

    for (int elem: evensIn(values)) {
        cout << elem << endl;
    }

    return 0;
}
```

Your Action Items

- Read Chapter 1 of *Programming Abstractions in C++* to learn more about the basics of C++ programming.
  - If you're coming from Java or JavaScript, much of this syntax will seem familiar, but there are some notable differences.
  - If you're coming from Python, it's pretty similar, but with lots of curly braces and semicolons.
- We'll begin writing C++ code in earnest on Wednesday.

Your Action Items

- ***Assignment 0: Welcome to CS106B*** is due this Friday at the start of class (11:30AM).
  - Starter files and assignment handout are up on the course website.
  - No programming involved, but you'll need to get your development environment set up.
- There's a bunch of documentation up on the course website. Please feel free to reach out to us if there's anything we can do to help out!

Your Action Items

- Some of the later assignments can be done in pairs.
  - Assignment 0 must be done individually. Everyone needs to have a working development environment and know how to work the debugger.
  - You may want to start thinking about who you'd like to work with, since you'll need to register for the same section as the person you'll be working with.

Next Time

- ***Welcome to C++!***
  - Defining functions.
  - Reference parameters.
  - Introduction to recursion.