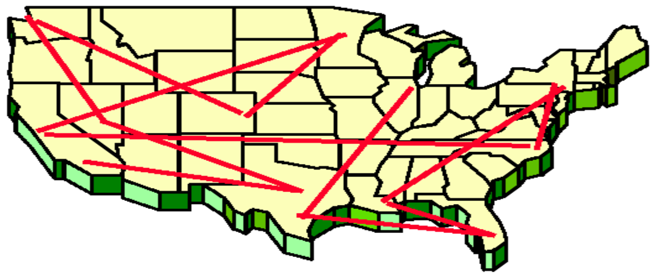
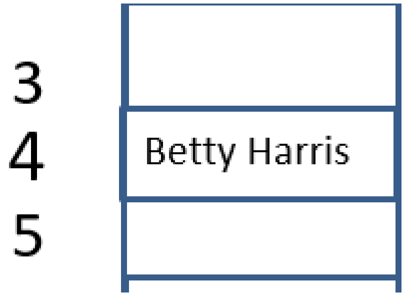
Compsci 101
Recursion and Beyond
Last Lecture

Susan Rodger
Alex Steiger
December 4, 2025





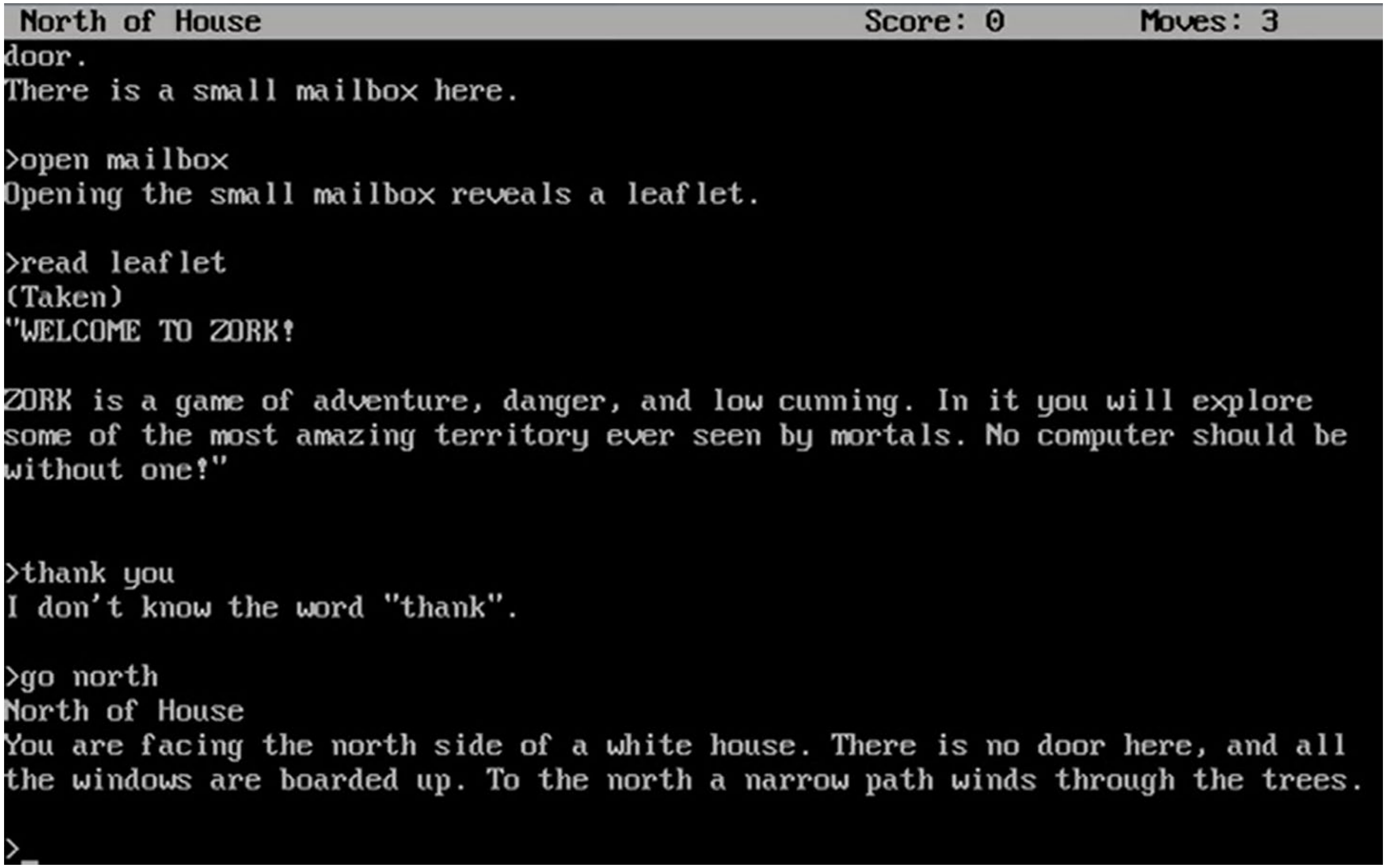
Y is for …

- **YAML and YACC**
  - Yet Another ...
- **Y2K:** https://www.youtube.com/watch?v=rblt2EtFfC4
  - How many bits are enough bits?
- **YouTube**
  - Connected to computing ...

Z is for …

- **Zero**
  - There are two, or 10 bits in the universe
- **Zip**
  - Compressed file archive format
- **Zork**
  - Text-based adventure game
  - https://www.youtube.com/watch?v=TNN4VPIRBJ8

Zork – Adventure game released in 1977



```
 North of House                                        Score: 0        Moves: 3
door.
There is a small mailbox here.

>open mailbox
Opening the small mailbox reveals a leaflet.

>read leaflet
(Taken)
"WELCOME TO ZORK!

ZORK is a game of adventure, danger, and low cunning. In it you will explore
some of the most amazing territory ever seen by mortals. No computer should be
without one!"


>thank you
I don't know the word "thank".

>go north
North of House
You are facing the north side of a white house. There is no door here, and all
the windows are boarded up. To the north a narrow path winds through the trees.

>_
```

Raja Kushalnagar

- Professor Gallaudet University
- PhD CS and MS of Laws (LLM) in Intellectual Property and Information Law from Univ Houston
- Juris Doctor (JD) Texas Southern University
- As a Deaf professor, he advocates in bringing consumers, industry, and policymakers together on accessibility issues, advocating for a deaf/hard of hearing perspective, as well as developing prototype technologies for improving the accessibility of such systems.

Announcements

- **APT-7 due today**
  - One grace day, NO LATE DAYS!
  - MUST TURN in BY tomorrow
- **Assign 8 Create due, tomorrow**
  - Extended Grace period is through Tues., Dec 9
  - No turnin's after that.
  - You turn in on Canvas
- **Last time for UTA office hours is tonight!**
  - Both Profs will have office hours posted on Ed
- **Exam 3 …. Regrade requests thru Wednesday night**

PFTD

- Final Exam
- How do dictionaries work so fast?
- Finishing up
  - What more is there in CS?

Final Exam –
Thurs, Dec 11
7pm-10pm

- Final Exam is in French Science 2231 for both sections of CompSci 101

- Set three alarms to wake up in time!!!!!!

Final Exam

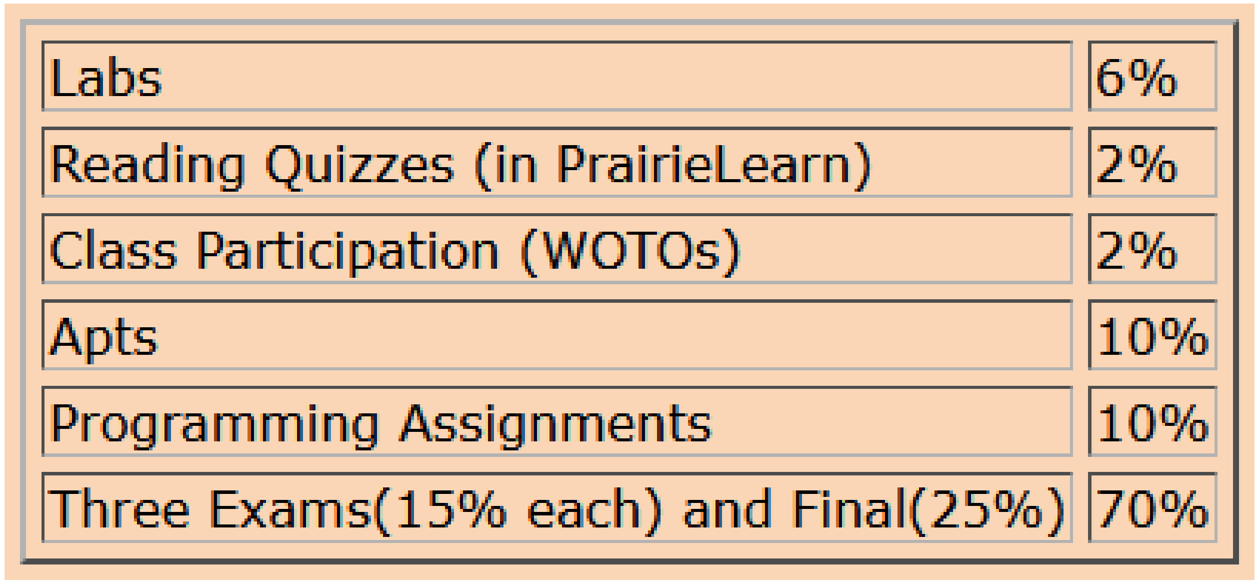
- **Study like you studied for Exam 3**
  - Use Exam 3 handout
- **We only have a little material since then**
  - Recommender assignment
    - This is all about stuff we did earlier
  - New topics, reading-level only:
    - Modules
    - Exceptions
    - Recursion
- **Not on the exam**
  - Images, turtles

IGNORE Grade Calculations on Canvas

- Your final grade is NOT calculated on Canvas!

Calculate Your Grade

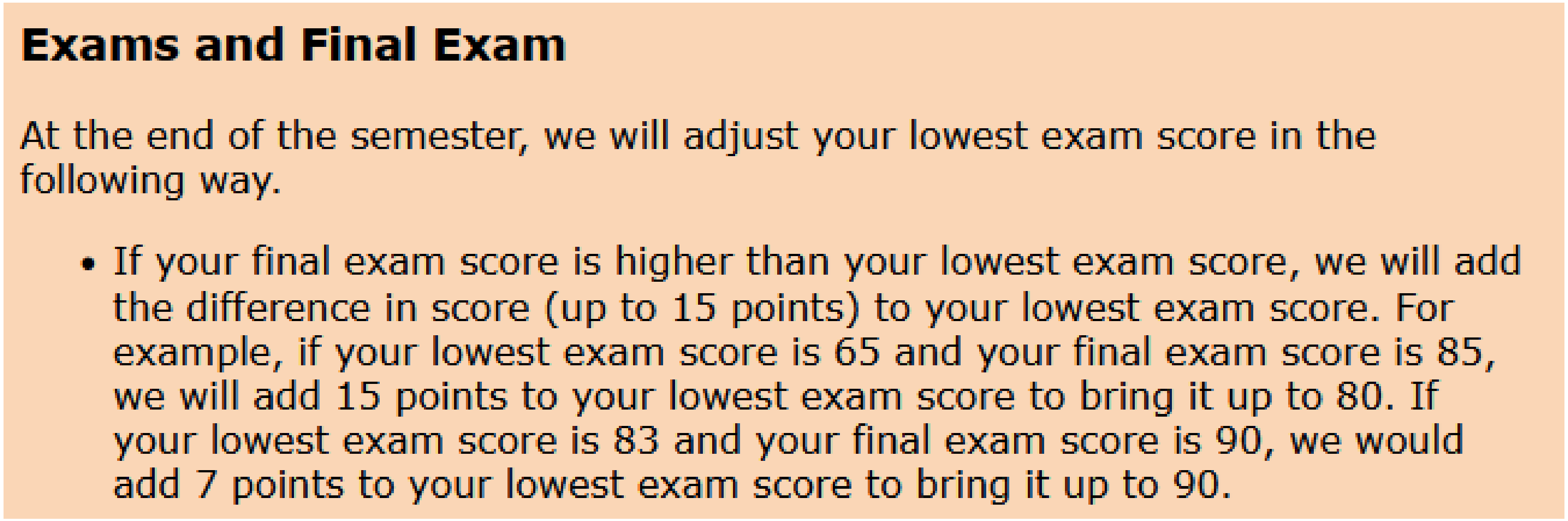
- From "About" tab on course web page



Labs	6%
Reading Quizzes (in PrairieLearn)	2%
Class Participation (WOTOs)	2%
Apts	10%
Programming Assignments	10%
Three Exams(15% each) and Final(25%)	70%

- Final exam is required

Will Calculate your final grade two ways! From webpage:



Exams and Final Exam

At the end of the semester, we will adjust your lowest exam score in the following way.

- If your final exam score is higher than your lowest exam score, we will add the difference in score (up to 15 points) to your lowest exam score. For example, if your lowest exam score is 65 and your final exam score is 85, we will add 15 points to your lowest exam score to bring it up to 80. If your lowest exam score is 83 and your final exam score is 90, we would add 7 points to your lowest exam score to bring it up to 90.

More on Grades

- Class Participation-WOTOs – drop 20 **points**
- PrairieLearn Quizzes – drop 4 **lowest** (of 27)
  - **`sum(sorted(scoresOutOf100)[4:])`**
- Lab – drop **15** points (each lab is 5 pts)

- That is all we drop

Profs Rodger/Steiger Office hours

- UTAs no longer have office hours after today

- Profs have office some hours next week
- Posted on Ed Discussion

How do Dictionaries work so fast?

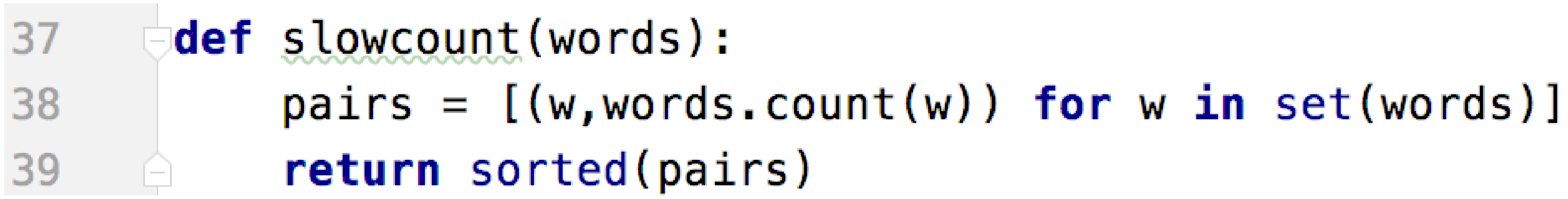
- How are they implemented?

Review: Problem – which word occurs the most frequently in a file

- Need to count how many times each word occurs
  - slowcount – for each word in a set, calls count
  - fastcount – counting dictionary

slowcount function
Short Code and Long Time

- See module WordFrequencies.py
  - Find # times each word in a list of words occurs
  - We have tuple/pair: word and word-frequency

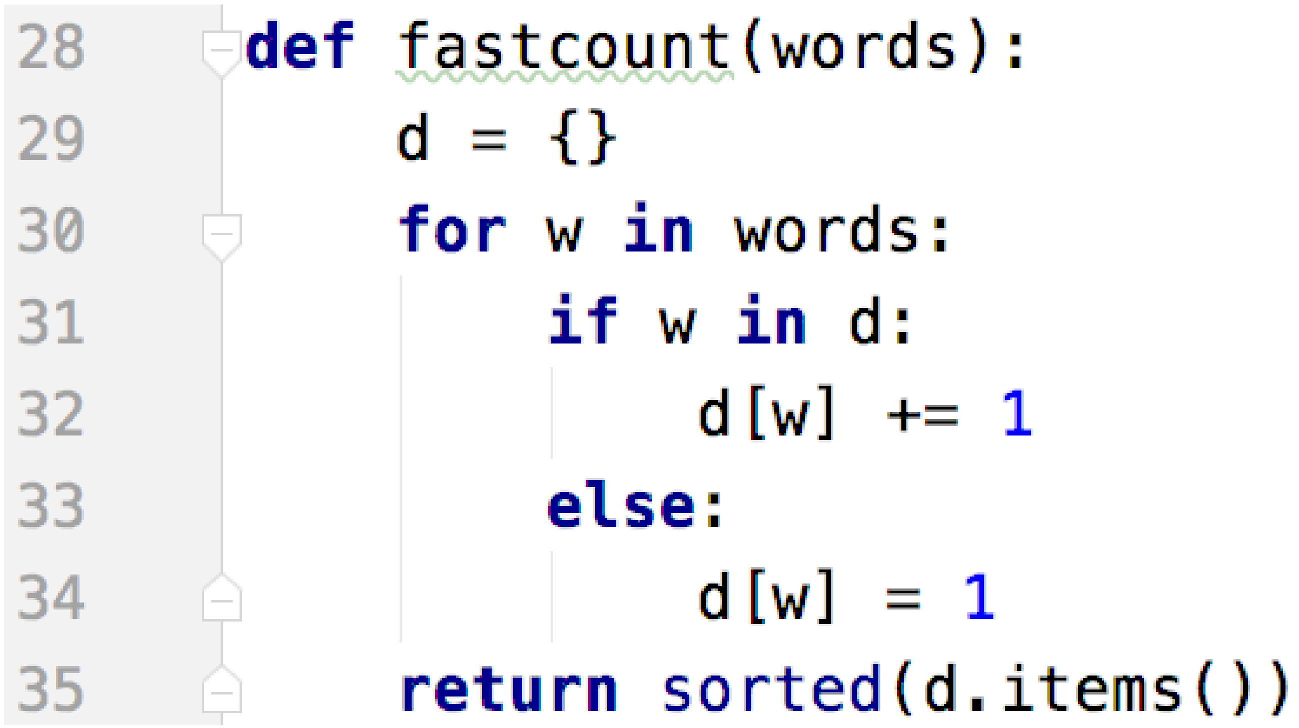


```
def slowcount(words):
    pairs = [(w,words.count(w)) for w in set(words)]
    return sorted(pairs)
```

- Think: How many times is **`words.count(w)`** called?
  - Why is **`set(words)`** used in list comprehension?

Using fastcount

- Update count if we've seen word before
  - Otherwise it's the first time, occurs once



```python
def fastcount(words):
    d = {}
    for w in words:
        if w in d:
            d[w] += 1
        else:
            d[w] = 1
    return sorted(d.items())
```

Let’s run them and compare them!

- **Run with Melville and observe time**
  - slowcount about 0.76 seconds
  - fastcount about 0.00 seconds

- **Run with Hawthorne and observe time**
  - slowcount about 14.6 seconds
  - fastcount about 0.03 seconds

Here is the idea behind how dictionaries work

Simple Example

Want a mapping of Soc Sec Num to Names

- Duke's CS Student Union wants to be able to quickly find out info about its members. Also add, delete and update members. Doesn't need members sorted.

267-89-5431	John Smith
703-25-6141	Ademola Olayinka
319-86-2115	Betty Harris
476-82-5120	Rose Black

- Dictionary d – SSN to names
  - d['267-89-5431'] = 'John Smith'
  - How does it find 'John Smith' so fast?

Dictionary d(SSN) = (SSN, name)

- We actually would map the SSN to the tuple of (SSN, name).
- That is a lot to display on a slide, so we will just show SSN to name
- But remember name is really a tuple of (SSN,name)

Simple Example

Let's look under the hood.
How are dictionaries implemented?

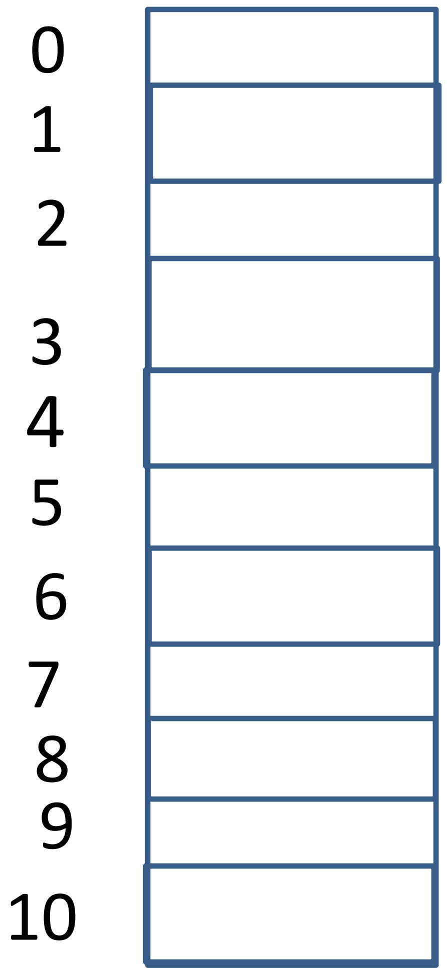
- Dictionaries implemented with a list, in a clever way
- How do we put something into the list fast?
- How do we find it in the list quickly?
  - d['267-89-5431'] = 'John Smith'
- List size is 11 – from 0 to 10
- d['267-89-5431'] calculates index in list of where to put 'John Smith' with hash function H:
  - H(SSN) = (last 2 digits of SSN) mod 11

Have a list of size 11 from 0 to 10

- Insert these into the list
- Insert as (key, value) tuple

  (267-89-5431, John Smith)

  (in example, only showing name)



Time for CompSci 101 Course Eval

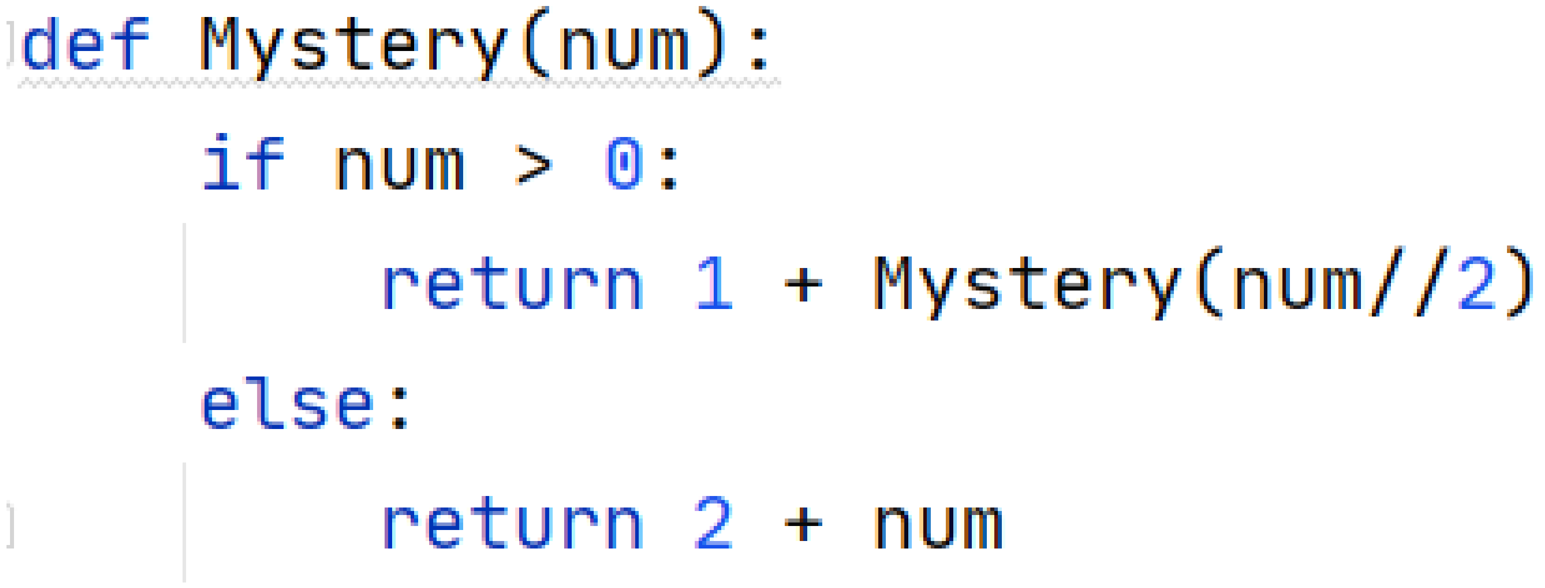
Please fill out Duke Course Eval for the Lecture and give us feedback on the course!

- 10:05 section is Lect Sec 01
- 3:05pm section is Lect Sec 02

1. If we get 65% filled out – everyone gets 1 extra point on Exam 3
2. If we get 75% filled out – everyone gets 1 more extra point on Exam 3

Review Recursion

Mystery Recursion



```python
def Mystery(num):
    if num > 0:
        return 1 + Mystery(num//2)
    else:
        return 2 + num
```

A peek into CompSci 201

- **Sorting list of numbers**
- **Quicksort algorithm**
    - Uses recursion

Quicksort - Idea

- Pivot – select and adjust the list
  - Select one of the elements
  - Put it where it belongs in sorted order
  - Put elements less than it, to its left
  - Put elements greater than it, to its right
- Recursively sort the elements to its left
- Recursively sort the elements to its right
- Done!

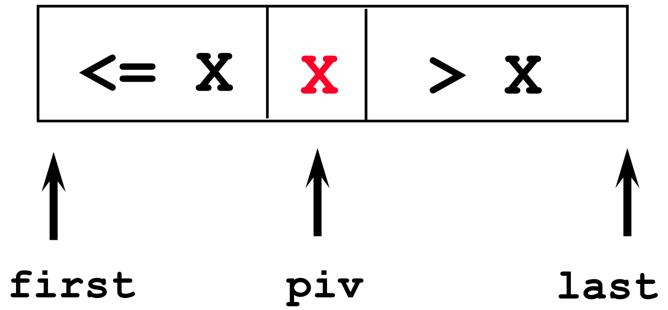
Quicksort - Idea

- Pivot – select and adjust list
- Recursively sort the elements to its left
- Recursively sort the elements to its right

5 9 1 4 3 6 2 7

Quicksort: fast in practice

```
def doQuick(list, first, last) {
    if (first >= last) return

    piv = pivot(list,first,last)
    doQuick(list,first,piv-1)
    doQuick(list,piv+1,last)
}
```

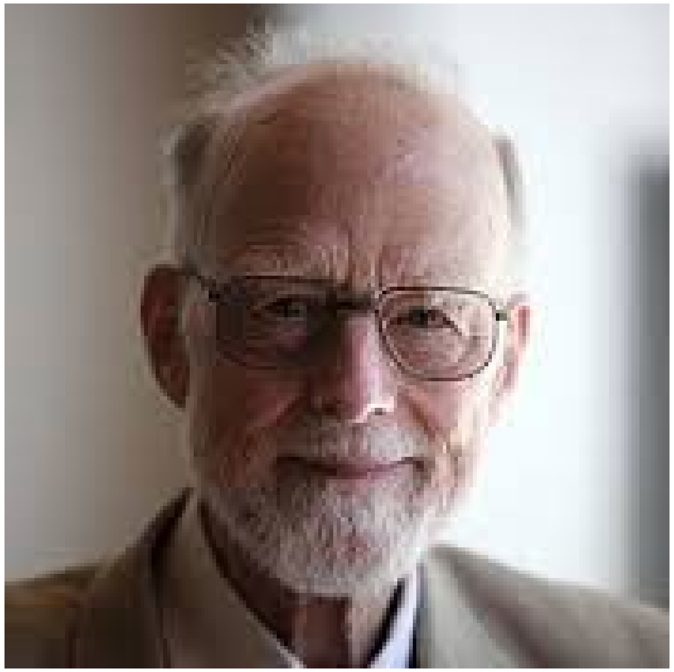


Sir Anthony (Tony) Hoare

Invented Quicksort in 1962
- he didn't get recursion

Turing Award winner
- programming language design
- Algol 60

"There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. And the other way is to make it so complicated that there are no obvious deficiencies."



"Inside every large program is a small program struggling to get out."

What is Computing? Informatics?

- **What is computer science, what is its potential?**
  - What can we do with computers in our lives?
  - What can we do with computing for society?
  - Will networks transform thinking/knowing/doing?
  - Society affecting and affected by computing?
  - Changes in science: biology, physics, chemistry, …
  - Changes in humanity: access, revolution (?), …

- **Privileges and opportunities available if you know code**
  - Writing and reading code, understanding algorithms
  - Majestic, magical, mathematical, mysterious, …

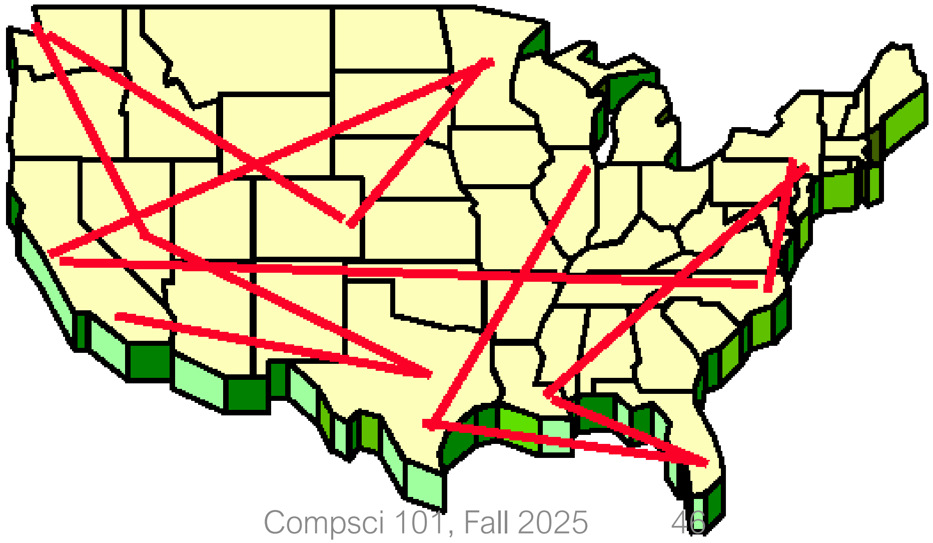
Computing - solve all problems?

- **Some problems can be solved 'efficiently'**
  - Run large versions fast on modern computers
  - What is 'efficient'? It depends
- **Some cannot be solved by computer.**
  - Provable! We can't wait for smarter algorithms

- **Some problems have no efficient solution**
  - Provably exponential 2^n so for "small" n …
- **Some have no known efficient solution, but**
  - If one does they all do!

Problem: Traveling Band

- Band wants you to schedule their concerts.
- They don’t like to travel. Minimize the time they are on the bus!
- Given N cities, what is the best schedule (shortest distance) to visit all N cities once?



How do you calculate the best path?

- Try all paths
  - Atlanta, Raleigh, Dallas, Reno, Chicago
    - Add up the distance in this order
  - Dallas, Atlanta, Raleigh, Reno, Chicago
    - Add up the distance in this order
  - Etc.
- – Find the min length path
- Would you agree to code this up?

How is Python like all other programming languages, how is it different?

Find all unique/different words in a file, in sorted order

Unique Words in Python

```
def main():
    f = open('/data/melville.txt', 'r')
    words = f.read().split()
    allWords = set(words)

    for word in sorted(allWords):
        print(word)

if __name__ == "__main__":
    main()
```

Unique words in Java

```
import java.util.*;
import java.io.*;
public class Unique {
  public static void main(String[] args) throws
   IOException {
    Scanner scan = new Scanner(
                   new File("/data/melville.txt"));
    TreeSet<String> set = new TreeSet<String>();

    while (scan.hasNext()){
        String str = scan.next();
        set.add(str);
    }
    for(String s : set){
        System.out.println(s);
    }
  }
}
```

Unique words in C++

```
#include <iostream>
#include <fstream>
#include <set>
using namespace std;

int main(){
  ifstream input("/data/melville.txt");
  set<string> unique;
  string word;
  while (input >> word){
    unique.insert(word);
  }
  set<string>::iterator it = unique.begin();
  for(; it != unique.end(); it++){
    cout << *it << endl;
  }
  return 0;
}
```

Unique words in PHP

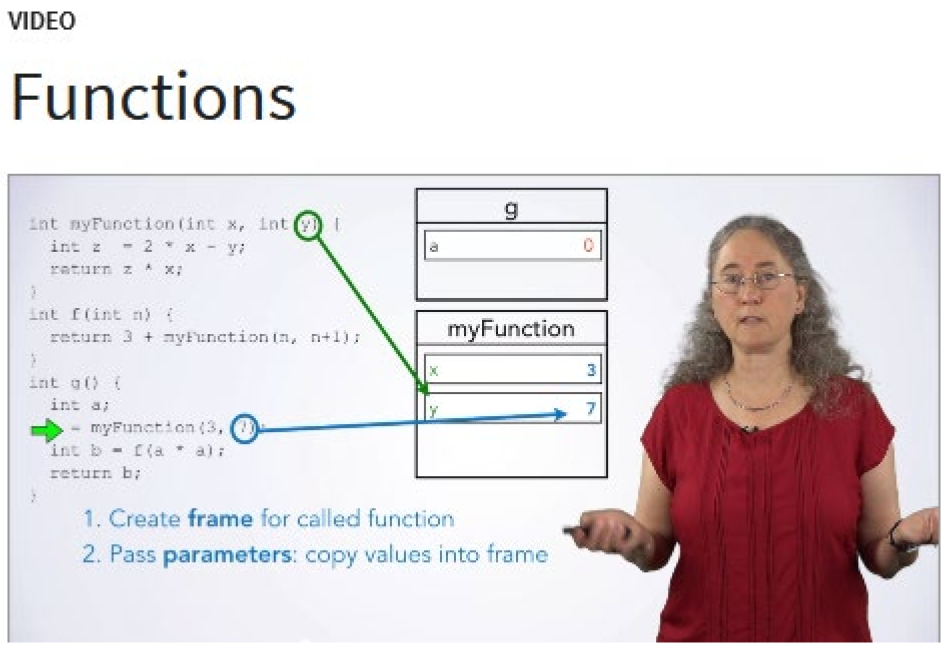
```
<?php

$wholething =
   file_get_contents("file:///data/melville.txt");
$wholething = trim($wholething);

$array = preg_split("/\s+/",$wholething);
$uni = array_unique($array);
sort($uni);
foreach ($uni as $word){
   echo $word."<br>";
}

?>
```

What is next?

- CompSci 201
  - Java, efficiency, other ways to organize data
- CompSci 230 or 231 or 232 – can take concurrently with 201
  - Discrete Mathematics (three different versions)
    - Course substitutions if you take a lot of math/stats
- CompSci 260 Computational Biology
- CompSci 216 Everything Data
- CompSci 240 Race, Gender, Class and Computing

What to do over break?
Take a Duke Coursera course Free

- Course on Java



VIDEO

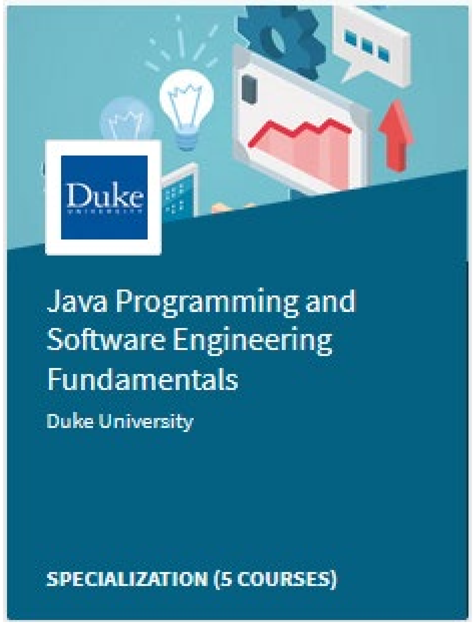
Functions

```
int myFunction(int x, int y) {
  int z  = 2 * x - y;
  return z * x;
}
int f(int n) {
  return 3 + myFunction(n, n+1);
}
int g() {
  int a;
  a = myFunction(3, 7);
  int b = f(a * a);
  return b;
}
```

1. Create **frame** for called function
2. Pass **parameters**: copy values into frame

Duke Coursera course on Java

- Coursera for Duke
  - https://online.duke.edu/coursera-for-duke/
- Java Programming/Soft Eng Fund
  - 5 courses
    - HTML/CSS/JavaScript
    - 4 courses on Java
- Course is FREE for Duke students
  - Go through link above


Duke
Java Programming and Software Engineering Fundamentals
Duke University
SPECIALIZATION (5 COURSES)