Plan for LDO101

- Ethical webpage scraping
  - Illustrate power of regular expressions
  - Python makes trying things relatively easy

- What's left, grades, finals, work
  - Optional APT, lab, finishing, studying
- What can't be done in Computer Science
  - Practical knowledge of theoretical concepts
- Acknowledging Completion

Grading

- There are 11 labs, each worth 4 points
  - Will grade with max of 38 points needed, 10%
- Forty APTs are required (53 given)
  - Grades for 41 in Sakai, missing 8-10, 10%
- Reading quizzes, we drop 20 points
  - Class activity will update and drop 4 points
- For any concerns, fill out form by 12/4
  - http://bit.ly/101fall15-concern

Final Exam

- Material from semester, emphasizes recent material, builds on all
  - Coding questions like midterm exams
- Multiple choice questions similar to in-class,
  - We have to grade these quickly
- Best study? Look at previous midterms, be able to do our last midterm

Be a UTA!! Help next semester's 101

http://bit.ly/compsci-uta

Ignorable: Dictionary Comprehensions

- Given `["x", "y", "z", "w"]` create dictionary for each element, value is empty list

```
d = {}
for val in letters:
    d[val] = []
```

- Use dictionary comprehension

```
d = {elt:[] for elt in letters}
```

- Initializes dictionary, just update

```
for elt in letters:
    d[elt].append(word.find(elt))
```

FriendScore APT

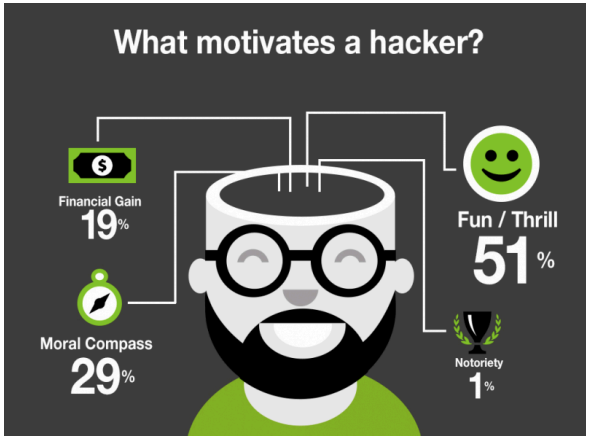
- What is a two-friend, doing an example by hand paper-and-pencil
  - How do we find indexes of our friends?
  - How could we find indexes of another person's friends?
  - If Sam is my friend, and Pat is Sam's friend, is Pat my two-friend? Is Pat's friend Chris my 2F?
- Try in-class questions toward going green
  - http://bit.ly/101fall15-dec3-1

Answer Questions

http://bit.ly/101fall15-dec3-1

Scraping email address from websites

- Suppose we want to send email to all Duke Faculty to let them know …
  - Visit Departmental website, people, faculty
  - View (HTML) Source
  - Develop regex to access email – if possible!
- RegexScraper.py
  - Python makes this simple
  - Ethical hacking?



Scraping math.duke.edu faculty

- **Pattern:**
  - `r'math/faculty/(.*?)\"\>(.+?)\<'`
- **URL**
  - http://fds.duke.edu/db/aas/math/faculty/
- **Matches:**

```
…
('motta', 'Francis C. Motta')
('jmmza', 'James Murphy')
('ryser', 'Marc D. Ryser')
('sv113', 'Stefano Vigogna')
('haizhao', 'Haizhao Yang')
```

Scraping Sanford/PubPol faculty

- **Pattern:**
  - `r'(\w+[.\w]*)@(\w+[.\w+]*)'`
- **URL**
  - https://sanford.duke.edu/people…/
- **Matches (call 16 times with different URL)**

```
…
('schanzer', 'duke.edu')
('steveschewel', 'gmail.com')
('michael.schoenfeld', 'duke.edu')
('schroeder', 'law.duke.edu')
```

Scraping Biology faculty

- **Pattern:**
  - `r'mailto:(\w+[.\w]*)@(\w+[.\w+]*)'`
- **URL**
  - https://biology.duke.edu/people/all-faculty/a
- **Matches (call 26 times with different URL)**

```
…
('emily.bernhardt', 'duke.edu')
('emily.bernhardt', 'duke.edu')
('bhandawat', 'gmail.com')
('bhandawat', 'gmail.com')
('jboynton66', 'gmail.com')
('jboynton66', 'gmail.com')
```

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, ...

What can be programmed?

- **What class of problems can be *solved*?**
  - **Linux, Cloud, Mac, Windows10, Android,…**
  - **Alan Turing contributions**
    - **Halting problem, Church-Turing thesis**

- **What class of problems can be *solved efficiently*?**
  - **Problems with no practical solution**
  - **What does practical mean?**

Schedule students, minimize conflicts

- Given student requests, available teachers
  - write a program that schedules classes
  - Minimize conflicts
- Add a GUI too
  - Web interface
  - …
  - …

Still another scenario, is this better?

Summary of Problem Categories

- 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!

Entscheidungsproblem

- What can we program?
  - What kind of computer?
- What can't we program?
  - Can't we try harder?
- Can we write a program that will determine if any program *P* will halt when run on input *S*?
  - Input to halt: P and S
  - Output: yes/no halts

Some problems take forever, but …

- Can we visit all cities, no repeats, using Southwest, for less than $123,329.50
  - RDU->MCO->…->…->…->…->DEN
  - RDU->DEN->…->…->…->…->MCO
  - repeat and test, what's the issue here?
  - Can we find shortest path for packets on Internet? Yes!
  - Can we find longest path for silent meditation? No!
  - We don't know how, but if we did!!!

- Contrast towers of Hanoi, 2^n moves always!

Are hard problems easy? Clay Prize

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

A Rose by any other name…C or Java?

- Why do we use [Python | Java] in courses ?
  - [is | is not] Object oriented
  - Large collection of libraries
  - Safe for advanced programming and beginners
  - Harder to shoot ourselves in the foot
- Why don't we use C++ (or C)?
  - Standard libraries weak or non-existant (comparatively)
  - Easy to make mistakes when beginning
  - No GUIs, complicated compilation model
  - What about other languages?

Why do we learn other languages?

- **Perl, Python, PHP, Ruby, C, C++, Java, Scheme, Haskell,**
  - Can we do something different in one language?
    - In theory: no; in practice: yes
  - What languages do you know? All of them.
  - In what languages are you fluent? None of them

- **In later courses why do we use C or C++?**
  - Closer to the machine, understand abstractions at many levels

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

Across different languages: do these languages have the same power?

Unique Words in Python

```
def main():
    f = open('/data/melville.txt', 'r')
    words = f.read().strip().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++

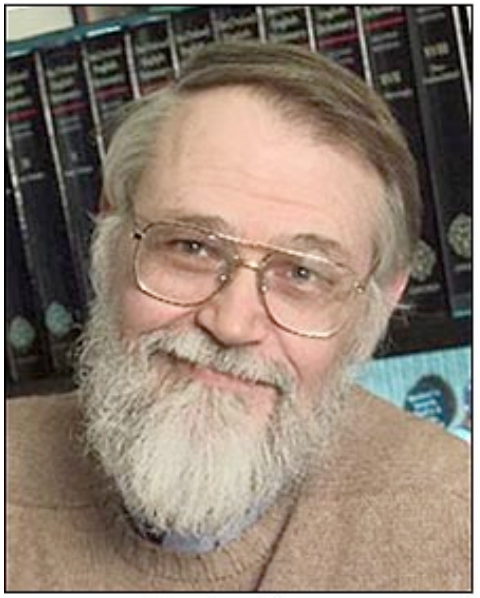
```
#include <iostream>
#include <fstream>
#include <set>
using namespace std;

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;
}
```

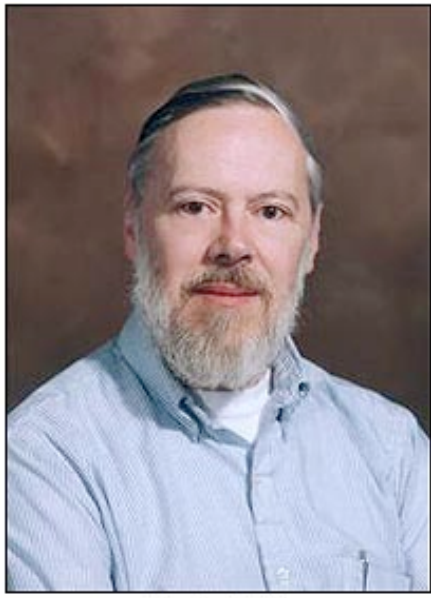
Unique words in PHP

```
<?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>";
}

?>
```

Kernighan and Ritchie

- First C book, 1978
- First 'hello world'
- Ritchie: Unix too!
  - Turing award 1983
- Kernighan: tools
  - Strunk and White


Brian Kernighan


Dennis Ritchie

- Everyone knows that debugging is twice as hard as writing a program in the first place. So if you are as clever as you can be when you write it, how will you ever debug it?

Brian Kernighan

How do we read a file in C?

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int strcompare(const void * a, const void * b){
  char ** stra = (char **) a;
  char ** strb = (char **) b;
  return strcmp(*stra, *strb);
}

int main(){
  FILE * file = fopen("/data/melville.txt","r");
  char buf[1024];
  char ** words = (char **) malloc(5000*sizeof(char **));
  int count = 0;
  int k;
```

Storing words read when reading in C

```
while (fscanf(file,"%s",buf) != EOF){
  int found = 0;    // look for word just read
  for(k=0; k < count; k++){
    if (strcmp(buf,words[k]) == 0){
      found = 1;
      break;
    }
  }
  if (!found){      // not found, add to list
    words[count] = (char *) malloc(strlen(buf)+1);
    strcpy(words[count],buf);
    count++;
  }
}
```

- Complexity of reading/storing? Allocation of memory?

Sorting, Printing, Freeing in C

```
  qsort(words,count,sizeof(char *), strcompare);
  for(k=0; k < count; k++) {
    printf("%s\n",words[k]);
  }

  for(k=0; k < count; k++){
    free(words[k]);
  }
  free(words);

}
```

- Sorting, printing, and freeing
  - Ugh!

You have (almost) finished Compsci 101

- Let's talk about next steps and finishing this semester