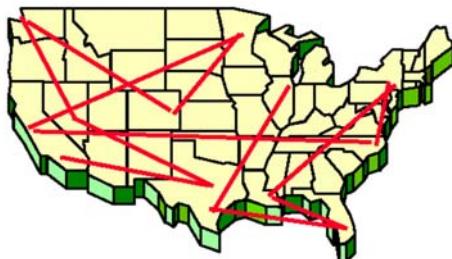


CompSci 101

Introduction to Computer Science



Dec 8, 2016

Prof. Rodger

compsci 101 fall 2016

1

Announcements

- Last Day of class!
- Assign 8 – With LATE Penalty thru Fri, Dec 9
- Assign 9 by Monday, none accepted after that
- APT 10 due by Monday, no Late APTs
- Form for taking Final exam another time
 - accommodations?
 - Three exams in a 24 hour period?
 - Room for some to take final with the other section
 - Fill out by tomorrow for consideration!!!

2

More Announcements

- Regrade for Exam 2 – submit by Friday, Dec 9
- Review Session – Tuesday 4pm (LSRC B101)
- Last Consulting Hours tonight
- Prof. Rodger will have office hours
 - Today 3-5pm, Tomorrow 2-5pm, more...
- Concern form – last minute concerns
- Today:
 - Wrapping up, Beyond CompSci 101
 - The Final exam

compsci 101 fall 2016

3

Final Exam

- Sec 01 – Monday, Dec 19, 2pm, **LSRC B101**
- Sec 02 – Thurs, Dec 15, 7pm, **Bio Sci 111**
- Closed Book, Closed Notes, Closed neighbor
- Python Reference Sheet
- Covers all topics through today
- Best way to study is practice writing code!
- See old tests (no old final exams)

compsci 101 fall 2016

4

Final Exam (cont)

- Test format
 - Multiple choice
 - Writing code – similar to exam 2
- Topics include:
 - if, loops, lists, sets, dictionaries, files, functions, sorting, etc
 - recursion, regular expressions – reading level only

compsci 101 fall 2016

5

Calculate Your Grade

- From “About” tab on course web page

Labs	10%
Reading Quizzes	5%
Class/Group work	5%
Apts	10%
Programming Assignments	10%
APT Quizzes	5%
Two Midterm Exams	30%
final exam	25%

compsci 101 fall 2016

6

More on Grades

- Lecture – ignore the first two weeks (drop/add period), plus drop 4 points
- Reading Quizzes – will drop 30 points
 - Lots of problems with Sakai this semester
 - Check your grades to make sure they copied over – fill out duke oit help form if they are wrong
- Lab – drop 6 points (each lab is 4 pts)
 - 44 pts total– 38 pts is 100%

7

Fill out Duke Course Eval

- Please fill out Duke Course Eval on DukeHub now
 - Only 15% have filled it in as of last night
- If you already have , then go to Sakai and fill out feedback on UTAs

compsci 101 fall 2016

8

Review - Selection Sort

- Sort a list of numbers.
- Idea:
 - Repeat til sorted
 - Find the smallest element in part of list not sorted
 - Put it where it belongs in sorted order.
 - Swap it with the element where it should be
- Sort example

<i>Sorted, won't move final position</i>	???
--	-----

compsci 101 fall 2016

9

Selection Sort – red area sorted

9	5	4	1	3	6	-	find smallest, swap
1	5	4	9	3	6	-	end of 1 st pass
1	5	4	9	3	6	-	find smallest, swap
1	3	4	9	5	6	-	end of 2 nd pass
1	3	4	9	5	6	-	find smallest, swap

10

Selection Sort (cont.)

1	3	4	9	5	6	-	end of 3 rd pass
1	3	4	9	5	6	-	find smallest, swap
1	3	4	5	9	6	-	end of 4th pass
1	3	4	5	9	6	-	find smallest, swap
1	3	4	5	6	9	-	end of 5 th pass, done

11

Bubble Sort

- Sort a list of numbers.
- Idea:
 - Repeat til sorted
 - Compare all adjacent pairs, one at a time. If out of order then swap them
- Sort example

???	<i>Sorted, won't move final position</i>
-----	--

compsci 101 fall 2016

12

Bubble Sort

bit.ly/101f16-1208-1

- Sort the list of numbers using BubbleSort.
- The body of the loop is one pass.
- Show the elements after each pass.
- [9, 5, 4, 1, 3, 6]

compsci 101 fall 2016

13

Bubble Sort – red area sorted

9	5	4	1	3	6	-	compare, swap
5	9	4	1	3	6	-	compare, swap
5	4	9	1	3	6	-	compare, swap
5	4	1	9	3	6	-	compare, swap
5	4	1	3	9	6	-	compare, swap
5	4	1	3	6	9	-	end of 1 st pass
5	4	1	3	6	9		

compsci 101 fall 2016

14

Bubble Sort – red area sorted

5	4	1	3	6	9	-	compare, swap
4	5	1	3	6	9	-	compare, swap
4	1	5	3	6	9	-	compare, swap
4	1	3	5	6	9	-	compare, no swap
4	1	3	5	6	9	-	end of 2cd pass
4	1	3	5	6	9		
4	1	3	5	6	9		

compsci 101 fall 2016

15

Bubble Sort – red area sorted

4	1	3	5	6	9	-	compare, swap
1	4	3	5	6	9	-	compare, swap
1	3	4	5	6	9	-	compare, no swap
1	3	4	5	6	9	-	end of 3 rd pass
1	3	4	5	6	9		

compsci 101 fall 2016

16

Two more passes would guarantee sorted.
Or Check if sorted and skip last two passes

Code for Bubblesort

```
def bubblesort(data):
    for j in range(len(data)-1,0,-1):
        print data
        for k in range(0,j):
            if data[k] > data[k+1]:
                data[k],data[k+1] = data[k+1], data[k]
    return data
```

compsci 101 fall 2016

17

Insertion Sort

bit.ly/101f16-1208-2

- Sort the list of numbers using InsertionSort.
- The body of the loop is one pass.
- Show the elements after each pass.
- [9, 5, 1, 4, 3, 6]

compsci 101 fall 2016

19

Insertion Sort

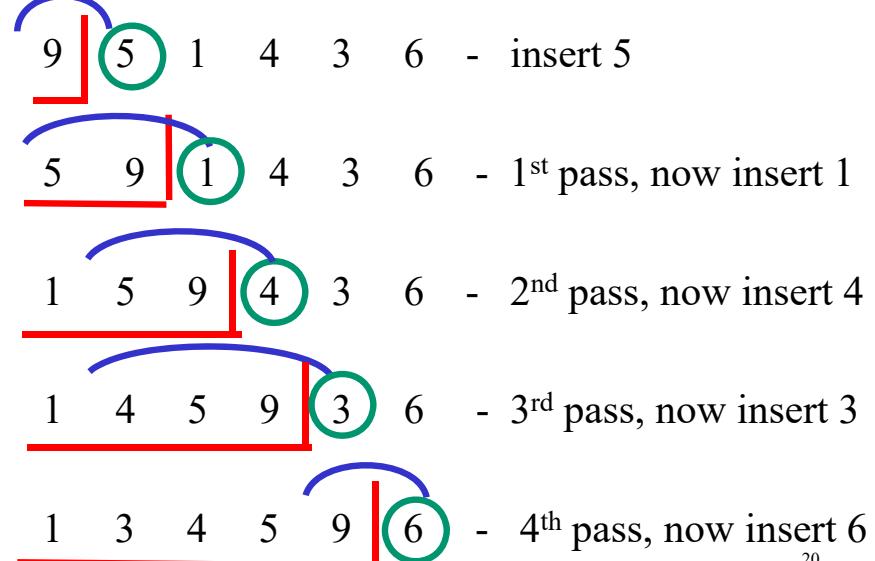
- Sort a list of numbers.
- Idea:
 - Sort by repeated inserting another element
 - Leftmost element is sorted part of list
 - Insert another element in that sublist keeping it sorted
 - Insert another element in that sublist keeping it sorted
 - Etc.
- Sort example

Sorted relative to each other	???
-------------------------------	-----

compsci 101 fall 2016

18

Insertion Sort – red area sorted



20

Insertion Sort – red area sorted

9 3 4 5 6 9 - 5th pass

21

Merge Sort

- Idea: Divide and Conquer
- Divide list into two halves
- Sort both halves (smaller problem)
- Merge the two sorted halves

9 5 1 4 3 6 2 7

compsci 101 fall 2016

22

Merge Sort

- Idea: Divide and Conquer
- Divide list into two halves
- Sort both halves (smaller problem)
- Merge the two sorted halves

9 5 1 4 3 6 2 7

9 5 1 4 3 6 2 7 divide list into 2 halves

compsci 101 fall 2016

23

Merge Sort

- Idea: Divide and Conquer
- Divide list into two halves
- Sort both halves (smaller problem)
- Merge the two sorted halves

9 5 1 4 3 6 2 7

9 5 1 4 3 6 2 7 divide list into 2 halves
1 4 5 9 2 3 6 7 recursively sort each half

compsci 101 fall 2016

24

Merge Sort

- Idea: Divide and Conquer
- Divide list into two halves
- Sort both halves (smaller problem)
- Merge the two sorted halves

9 5 1 4 3 6 2 7

9 5 1 4 3 6 2 7 divide list into 2 halves

1 4 5 9 2 3 6 7 recursively sort each half

1 2 3 4 5 6 7 9 merge the two sorted list

compsci 101 fall 2016

25

MergeSort idea for code

```
def mergesort(data)
    n = len(data)
    if n == 1:
        return data
    else:
        d1 = mergesort(data[:n/2])
        d2 = mergesort(data[n/2:])
        return merge(d1, d2)
```

26

What does recursively sort mean? Merge Sort

- Use the same Merge Sort algorithm
 - Divide list into two halves
 - Sort both halves (smaller problem)
 - Merge the two sorted halves

9 5 1 4

9 5 1 4 divide list into 2 halves

5 9 1 4 recursively sort each half

1 4 5 9 merge the two sorted list

compsci 101 fall 2016

26

bit.ly/101f16-1208-3

Question 1

Which sort is this?

4 10 5 3 8 2

4 10 5 3 8 2

4 5 10 3 8 2

3 4 5 10 8 2

3 4 5 8 10 2

2 3 4 5 8 10

Question 2

Which sort is this?

4 10 5 3 8 2

4 2 5 3 8 10

4 2 5 3 8 10

4 2 3 5 8 10

3 2 4 5 8 10

2 3 4 5 8 10

compsci 101 fall 2016

28

Wrap up Sorting

- Some Ways to Compare sorts.
 - How many total swaps?
 - Is one faster for certain types of input?
 - Does the input matter
- Different ways to sort?
 - Over 50 sorting algorithms
- Does President Obama know his sorts?
- Sorting animations
 - http://www.sorting-algorithms.com/



compsci 101 fall 2016

29

More on Sorting in CompSci 201

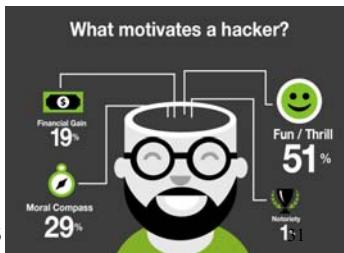
- Learn about this and other sorts in CompSci 201, also how to analyze them to determine which one works best.
- Python: Timsort
 - combines mergesort and insertion sort
- Shellsort
 - uses insertion sort on parts of the list repeatedly - those parts getting larger each time

compsci 101 fall 2016

30

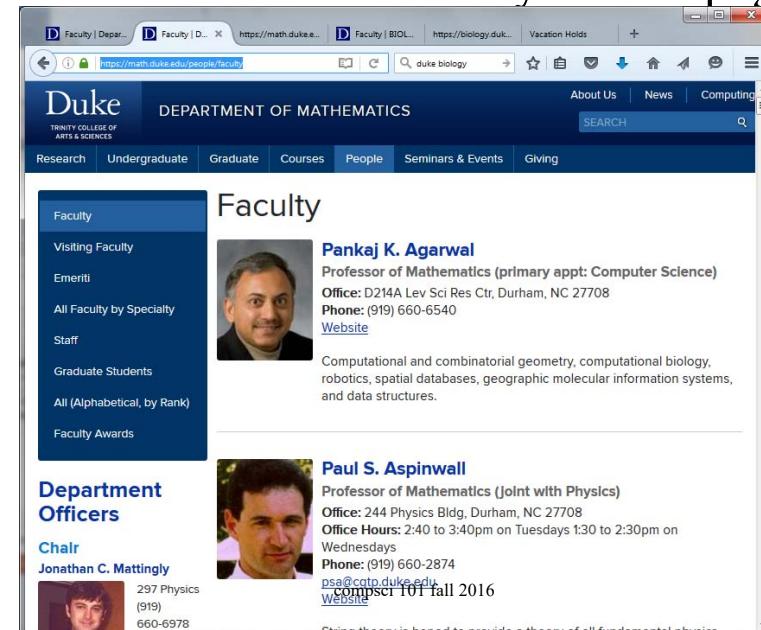
Scraping email address from websites

- 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?



compsci 101 fall 2016

Math Website – Faculty on one page



Pankaj K. Agarwal
Professor of Mathematics (primary appt: Computer Science)
Office: D214A Lev Sci Res Ctr, Durham, NC 27708
Phone: (919) 660-6540
[Website](#)

Computational and combinatorial geometry, computational biology, robotics, spatial databases, geographic molecular information systems, and data structures.

Paul S. Aspinwall
Professor of Mathematics (Joint with Physics)
Office: 244 Physics Bldg. Durham, NC 27708
Office Hours: 2:40 to 3:40pm on Tuesdays 1:30 to 2:30pm on Wednesdays
Phone: (919) 660-2874
[psa@math.duke.edu](#)
[Website](#)

32

Duke Biology Website A-Z pages

The screenshot shows the Duke Biology website's faculty search interface. The URL is <https://biology.duke.edu/people/all-faculty/a>. The page title is 'Faculty'. A search bar is at the top. Below it is a sidebar with links: 'All People', 'All Faculty', 'Graduate Faculty', 'Graduate Students', 'Postdocs', and 'Department Staff'. The main content area shows a grid of faculty profiles. One profile is highlighted for Susan C. Alberts, showing her photo, name, title (Robert F. Durden Professor of Biology), office information (Office: 130 Science Drive, Rm 137, Duke Box 90338, Durham, NC 27708, Campus Box 90338), phone (919) 660-7272, fax (919) 660-7293, and email (alberts@duke.edu). A link to her lab website (<http://www.biology.duke.edu/albertslab>) is also provided. Below the profile is a 'Full Profile' link and a note about her teaching: 'compsci 101 fall 2016'. At the bottom of the page is a link to 'Danielle Armaleo'.

33

View page source of html

The screenshot shows the raw HTML source code of the Duke Biology faculty page. A specific line containing a mailto: link for Susan C. Alberts is circled in red. The line is: `Full Profile`. The source code also includes other faculty profiles and their contact information.

34

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')

35

Public Policy pages for A-Z

The screenshot shows the Duke Sanford School of Public Policy's A-Z directory page. The URL is <https://sanford.duke.edu/people-and-research/directory/a>. The page title is 'A-Z Directory'. It features a grid of faculty profiles with columns for Name, Phone, and Email. The first few profiles listed are: Abels, Jonathan (Executive Director, Duke Center for International Development, Duke Center for International Development), Adair, Bill (Knight Professor of the Practice of Journalism and Public Policy, DeWitt Wallace Center for Media and Democracy), and Adler, Matthew D. (Richard A. Horvitz Professor of Law, Richard A. Horvitz Professor of Law).

36

Scraping Sanford/PubPol faculty

- Pattern:
 - `r' (\w+[.\w+]*)@(\w+[.\w+]*)'`
- URL
 - <https://sanford.duke.edu/people/>
- Matches (call 26 times with different URL)
 - ...
 - ('schanzer', 'duke.edu')
 - ('steveschewel', 'gmail.com')
 - ('michael.schoenfeld', 'duke.edu')
 - ('schroeder', 'law.duke.edu')

compsci 101 fall 2016

37

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, ...

compsci 101 fall 2016

38

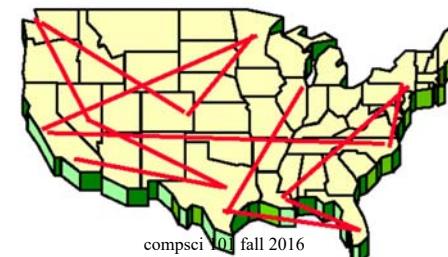
Computing - solve all problems?

- 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!

39

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?



compsci 101 fall 2016

40

How do you calculate the best path?

- Try all paths
 - Atlanta, Raleigh, Dallas, Reno, Chicago
 - Dallas, Atlanta, Raleigh, Reno, Chicago
 - Etc.
- Would you agree to code this up?

compsci 101 fall 2016

41

Answer questions
bit.ly/101f16-1208-4

compsci 101 fall 2016

42

How long?

Number of Cities	All paths – N!	Time to solve - 10^9 Instructions per second
10	3 million	
15	10^{12}	
18	10^{15}	
20	10^{18}	
25	10^{25}	

43

How long?

Number of Cities	All paths – N!	Time to solve - 10^9 Instructions per second
10	3 million	< sec
15	10^{12}	
18	10^{15}	
20	10^{18}	
25	10^{25}	

44

How long?

Number of Cities	All paths – $N!$	Time to solve - 10^9 Instructions per second
10	3 million	< sec
15	10^{12}	16 min
18	10^{15}	
20	10^{18}	
25	10^{25} compsci 101 fall 2016	45

How long?

Number of Cities	All paths – $N!$	Time to solve - 10^9 Instructions per second
10	3 million	< sec
15	10^{12}	16 min
18	10^{15}	11 days
20	10^{18}	
25	10^{25} compsci 101 fall 2016	46

How long?

Number of Cities	All paths – $N!$	Time to solve - 10^9 Instructions per second
10	3 million	< sec
15	10^{12}	16 min
18	10^{15}	11 days
20	10^{18}	31 years
25	10^{25} compsci 101 fall 2016	47

How long?

Number of Cities	All paths – $N!$	Time to solve - 10^9 Instructions per second
10	3 million	< sec
15	10^{12}	16 min
18	10^{15}	11 days
20	10^{18}	31 years
25	10^{25} compsci 101 fall 2016	10^8 years 48

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

compsci 101 fall 2016

49

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-existent (comparatively)
 - Easy to make mistakes when beginning
 - No GUIs, complicated compilation model
 - What about other languages?

compsci 101 fall 2016

50

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

compsci 101 fall 2016

51

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()
```

compsci 101 fall 2016

52

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);
    }
}
```

compsci 101 fall 2016

53

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;
}
```

compsci 101 fall 2016

54

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>";
}

?>
```

compsci 101 fall 2016

55

End with A CS Story bit.ly/101f16-1208-5

compsci 101 fall 2016

56