Evaluation

• Fill out course evaluation on ACES
• On Sakai (under announcements) please rate your Lab UTAs and any other UTAs you interacted with
Announcements

• Submit works now!
• Final Exam – accommodations/reschedule?
  – Fill out form by Dec 5
• APT 10 due Friday, last late day is Dec 7
• Asg 8 due tonight!, Last late day is Dec 6
• Asg 9 due Friday night, not accepted after midnight!
• Today
  – More on sorting, Classwork, CS story
More Announcements

• Be a UTA
  – http://www.cs.duke.edu/csed/uta/

• Next course
  – CompSci 201
  – Start all over again with Java
  – Java has if, loops, lists, maps (dictionaries), sets
  – Is that familiar?
  – Learn about nonlinear structures that can be more efficient
Final Exam

- Sec 01 (White Lect. Hall) – Sat Dec 13 2pm
- Sec 02 (LSRC B101) – Wed Dec 10 7pm
- 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)
- Fall 2014 tests – see other section tests
Final Exam (cont)

• Test format
  – Multiple choice
  – Writing code
• Topics include:
  – if, loops, lists, sets, maps, files, functions
  – recursion and regular expressions – reading level only
Calculate Your Grade

- From “About” tab on course web page

<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>labs</td>
<td>10%</td>
</tr>
<tr>
<td>quizzes (reading or knowledge) / classwork</td>
<td>10%</td>
</tr>
<tr>
<td>apts</td>
<td>10%</td>
</tr>
<tr>
<td>assignments</td>
<td>20%</td>
</tr>
<tr>
<td>two exams</td>
<td>25%</td>
</tr>
<tr>
<td>final exam</td>
<td>25%</td>
</tr>
</tbody>
</table>
More on Grades

• Lecture – drop the first two weeks (drop/add period) plus 3 more
• Reading Quizzes – will drop 20 points
• Lab – drop 8 points (each lab is 4 pts)
Wrap up Sorting

• Different ways to sort?
  – Over 50 sorting algorithms
• What sorting algorithm does Python sort use?
• Does President Obama know his sorts?
• Sorting animations
  http://www.sorting-algorithms.com/
Merge Sort

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

• Learn about this and other sorts in CompSci 201, also how to analyze them to determine which one works best.
• Timsort
• Shellsort
Classwork
bit.ly/101fall14-1204-01
Growth of functions

• As the size of the data increases, how many steps are there for an algorithm/method?
## Timings

<table>
<thead>
<tr>
<th>N</th>
<th>$\log_2 N$</th>
<th>$N^2$</th>
<th>$N^3$</th>
<th>$2^N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3.3</td>
<td>100</td>
<td>1000</td>
<td>1024</td>
</tr>
<tr>
<td>20</td>
<td>4.3</td>
<td>400</td>
<td>8000</td>
<td>1048576</td>
</tr>
<tr>
<td>40</td>
<td>5.3</td>
<td>1600</td>
<td>64000</td>
<td>$1.1 \times 10^{12}$</td>
</tr>
<tr>
<td>80</td>
<td>6.3</td>
<td>6400</td>
<td>512000</td>
<td>$1.2 \times 10^{24}$</td>
</tr>
<tr>
<td>160</td>
<td>7.3</td>
<td>25600</td>
<td>4096000</td>
<td>$1.4 \times 10^{48}$</td>
</tr>
</tbody>
</table>
## Timings

<table>
<thead>
<tr>
<th>N</th>
<th>$\log_2 N$</th>
<th>$N^2$</th>
<th>$N^3$</th>
<th>$2^N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>7.9</td>
<td>62,500</td>
<td>$1.56 \times 10^7$</td>
<td>$1.8 \times 10^{75}$</td>
</tr>
<tr>
<td>500</td>
<td>8.9</td>
<td>250,000</td>
<td>$1.25 \times 10^8$</td>
<td>$3.2 \times 10^{150}$</td>
</tr>
<tr>
<td>1000</td>
<td>9.9</td>
<td>$1 \times 10^6$</td>
<td>$1 \times 10^9$</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>10.9</td>
<td>$4 \times 10^6$</td>
<td>$4 \times 10^6$</td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>11.9</td>
<td>$1.6 \times 10^7$</td>
<td>$8 \times 10^9$</td>
<td></td>
</tr>
</tbody>
</table>
Look at the timings of the sorts

• How do the sorts compare?
  – With size as they grow
  – With different types of data
    • Random
    • Reverse
    • Almost sorted
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
  – Dallas, Atlanta, Raleigh, Reno, Chicago
  – Etc.

• Would you agree to code this up?
How long?

<table>
<thead>
<tr>
<th>Number of Cities</th>
<th>All paths – N!</th>
<th>Time to solve - $10^9$ Instructions per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3 million</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>$10^{12}$</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>$10^{15}$</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>$10^{18}$</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>$10^{25}$</td>
<td></td>
</tr>
</tbody>
</table>
P = NP?

- **P**: Problems with polynomial time solutions
  - \( N, N^2 \)
  - Example: Selection sort
  - Easy to solve

- **NP**: problems with not polynomial time solutions
  - \( 2^n, N! \)
  - Hard to solve
Does P = NP?

- Famous CS question
- If yes, a whole class of difficult problems can be solved efficiently, one problem is reducible to another
- If no, none of the hard problems can be solved efficiently
A CS Story