CompSci 101
Introduction to Computer Science

April 21, 2015
Prof. Rodger
Announcements

• Final Exam – accommodations/reschedule?
  – Fill out form by Wed. April 22!
• APT 10 due tonight, last late day is Apr 25
• Asg 9 due Thursday night, not accepted after midnight!
• Last Chance Concerns? - Fill out web form
• Prof. Rodger Office hours posted on Piazza
• Today
  – More on sorting, Classwork, Discuss Final,
  – 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
Finish Slides from Last Time

• Review

• Selectionsort
  • Each pass:
    • Select the next smallest
    • put where it belongs with one swap

• Bubblesort
  • Each pass
    • Compare adjacent pairs and swap if needed
    • Bubbles down largest element
Insertion Sort

• Sort a list of numbers.

• Idea:
  – Sort by repeatedly inserting another element
    • Leftmost part of list is sorted
    • Insert another element into the sorted part of the list
  – Leftmost sorted part of list starts with one element and adds one more element with each pass
  – Similar to holding playing cards in your hand sorted, and adding one more to the sorted hand

• Sort example
  
<table>
<thead>
<tr>
<th>Sorted relative to each other</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 1
bit.ly/101S15-0421

• 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]
Question 2
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 3
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
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
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
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
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
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
divide list into 2 halves
5 9 1 4 recursively sort each half
1 4 5 9 merge the two sorted list
Wrap up Sorting

• Some Ways to Compare these three sorts.
  • How many total swaps?
  • Is one faster for certain types of input?

• Different ways to sort?
  – Over 50 sorting algorithms

• What sorting algorithm does Python sort use?

• Sorting animations
  http://www.sorting-algorithms.com/
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.

- Timsort
  - combines mergesort and insertion sort

- Shellsort
  - uses insertion sort on parts of the list repeatedly - those parts getting larger each time
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 x 10^{12}</td>
</tr>
<tr>
<td>80</td>
<td>6.3</td>
<td>6400</td>
<td>512000</td>
<td>1.2 x 10^{24}</td>
</tr>
<tr>
<td>160</td>
<td>7.3</td>
<td>25600</td>
<td>4096000</td>
<td>1.4 x 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 x 10$^7$</td>
<td>1.8 x 10$^{75}$</td>
</tr>
<tr>
<td>500</td>
<td>8.9</td>
<td>250,000</td>
<td>1.25 x 10$^8$</td>
<td>3.2 x 10$^{150}$</td>
</tr>
<tr>
<td>1000</td>
<td>9.9</td>
<td>1 x 10$^6$</td>
<td>1 x 10$^9$</td>
<td>Error</td>
</tr>
<tr>
<td>2000</td>
<td>10.9</td>
<td>4 x 10$^6$</td>
<td>8 x 10$^9$</td>
<td>Error</td>
</tr>
<tr>
<td>4000</td>
<td>11.9</td>
<td>1.6 x 10$^7$</td>
<td>6.4 x 10$^{10}$</td>
<td>Error</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
Pause for Evaluation

• Fill out course evaluation on ACES

• On Sakai (under announcements) please rate your Lab UTAs and any other UTAs you interacted with
Final Exam

- Sec 01 (White Lect. Hall) – Fri May 1, 2pm
- Sec 02 (LSRC B101) – Mon Apr 27, 9am
- 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)
Final Exam (cont)

• Test format
  – Multiple choice
  – Writing code

• Topics include:
  – if, loops, lists, sets, maps, files, functions
  – recursion, regular expressions, sorting – 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>15%</td>
</tr>
<tr>
<td>assignments</td>
<td>15%</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 – ignore the first two weeks (drop/add period) plus drop 4 points
• Reading Quizzes – will drop 20 points
• Lab – drop 6 points (each lab is 4 pts)
Extra Optional Practice Problems

- One of the UTAs has written up lots of practice problems
  - sample input with resulting output
  - a wide range of problems - easy to challenging
  - linked into today’s lecture
  - not required
Now more on CS topics ...
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>
### 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>&lt; sec</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>
## 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>&lt; sec</td>
</tr>
<tr>
<td>15</td>
<td>$10^{12}$</td>
<td>16 min</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>
# 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>&lt; sec</td>
</tr>
<tr>
<td>15</td>
<td>$10^{12}$</td>
<td>16 min</td>
</tr>
<tr>
<td>18</td>
<td>$10^{15}$</td>
<td>11 days</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>
## 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>&lt; sec</td>
</tr>
<tr>
<td>15</td>
<td>$10^{12}$</td>
<td>16 min</td>
</tr>
<tr>
<td>18</td>
<td>$10^{15}$</td>
<td>11 days</td>
</tr>
<tr>
<td>20</td>
<td>$10^{18}$</td>
<td>31 years</td>
</tr>
<tr>
<td>25</td>
<td>$10^{25}$</td>
<td></td>
</tr>
<tr>
<td>Number of Cities</td>
<td>All paths – N!</td>
<td>Time to solve - $10^9$ Instructions per second</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>3 million</td>
<td>&lt; sec</td>
</tr>
<tr>
<td>15</td>
<td>$10^{12}$</td>
<td>16 min</td>
</tr>
<tr>
<td>18</td>
<td>$10^{15}$</td>
<td>11 days</td>
</tr>
<tr>
<td>20</td>
<td>$10^{18}$</td>
<td>31 years</td>
</tr>
<tr>
<td>25</td>
<td>$10^{25}$</td>
<td>$10^8$ years</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
End with A CS Story