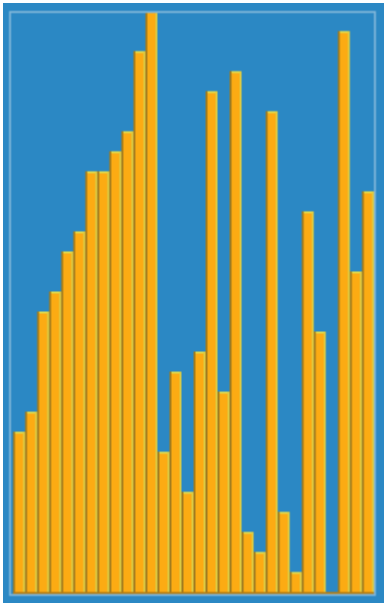


# CompSci 101

## Introduction to Computer Science



December 2, 2014

Prof. Rodger

# Announcements

- No Reading or RQ for next
- Assignments and APTs due this week!
  - Note Fewer late days!!!
  - Problem with Final Exam Date? Or have accommodations, must fill out form on course web page to reschedule exam
- Finish lecture notes from last time
- Today – Sorting how's

# Announcements (cont)

- Regrades for Exam 2
  - give to Prof. Rodger
- Be a UTA for CompSci 101
  - Rewarding and learning experience!

# Sorting

- In python:
  - `alist = [8, 5, 2, 3, 1, 6, 4]`
  - `alist.sort()`      OR      `result = sorted(alist)`
  - Now `alist` OR `result` is `[1, 2, 3, 4, 5, 6, 8]`
- How does one sort elements in order? How does “sort” work?

# Selection Sort

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

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

# Question 1 –

<http://bit.ly/101Fall14-1202-01>

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

# Question 2:

## Code for Selection Sort

- Snarf the code for today.
- Fill in the missing code for selection sort
- 1) First finish minIndex – returns the index of the minimum element in list items, between “start” and the right end of the list
- 2) Complete the body of the for loop in Selection sort

# 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>
-----	--

## Question 3 -

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

# Question 4:

## Code for Bubblesort

- Fill in the missing code for bubblesort
- 1) What is the range of the second for loop?
- 2) Complete the body of the 2nd for loop

# 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

<i>Sorted relative to each other</i>	???
--	-----

## Question 5 -

- 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 6:

## Code for InsertionSort

- Fill in the missing code for insertionsort
- 1) What are the conditions for the while?
- 2) Complete the body of the while

# Wrap up Sorting

- Question 7:
  - Compare these three sorts.
    - How are they the same?
    - How are they different?
- 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