Announcements

• RQ done!
• Assign 8 due today
• APT 10, Assign 9 – due Friday(Monday)
• Final Exam:
  – Sec 01 Mon Dec 19 2pm, LSRC B101
  – Sec 02 Thur Dec 15 7pm, BIO Sci 111
  – Get accommodations?
  – Room for some to take final with the other section
  – Must fill out form by THIS FRIDAY, Dec 9.

Calculate Your Grade

• From “About” tab on course web page

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labs</td>
<td>10%</td>
</tr>
<tr>
<td>Reading Quizzes</td>
<td>5%</td>
</tr>
<tr>
<td>Class/Group work</td>
<td>5%</td>
</tr>
<tr>
<td>Apts</td>
<td>10%</td>
</tr>
<tr>
<td>Programming Assignments</td>
<td>10%</td>
</tr>
<tr>
<td>APT Quizzes</td>
<td>5%</td>
</tr>
<tr>
<td>Two Midterm Exams</td>
<td>30%</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 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%
More Announcements

• Regrades for Exam 2 – submit by Fri. Dec 9
• Be a UTA for CompSci 101
  – Rewarding and Learning Experience
  – Apply: http://www.cs.duke.edu/csed/uta
• Last Lab this week

• Today:
  – More on Recursion, Regex
  – More on Sorting and analyzing it

Provide Comments on UTAs

• Lab UTAs
• Any other UTAs who helped you?

• See announcement in Sakai
  – Anonymous Feedback for course
  – Anonymous feedback on UTAs

Regex Questions
bit.ly/101f16-1206-1

Review Recursion and Regex
Dictionary Comprehension

- List comprehension - builds a new list
- Dictionary comprehension - builds a new dictionary

- Format
  
  \[ d = \{ \text{key: value for key in somelist if ...} \} \]

Example: From Exam 2 – dict of Actors to list of movies: (movie in, num minutes in)

```python
def dictActorsToMovies(data):
    d = {}
    for item in data:
        if item[1] not in d:
            d[item[1]] = [(item[0], item[4])]
        else:
            d[item[1]].append((item[0], item[4]))
    return d
```

Example: Assignment 8
ReadFood: Initialize dictionary ratingsdict

- Compute number of restaurants, say \( n \)
- Create \textbf{alldata} – list of
  
  \[[\text{name1}, \text{ratings1}], [\text{name2}, \text{ratings2}], [\text{name3}, \text{ratings3}], \ldots\]

  \[[\text{‘JoJo’}, [\text{‘Skillet’}, 1, \text{‘McDonalds’}, 1, \text{‘Tandoor’}, 3, \text{‘PandaExpress’}, 3]], \ldots\]

- Then create dictionary:
  
  \( \text{ratingsdict} = \{\text{person[0]}: [0]*n \text{ for person in somelist}\} \)

- Then update dictionary by processing \textbf{alldata}

Sorting

- In python:
  
  - \( \text{alist} = [8, 5, 2, 3, 1, 6, 4] \)
  
  - \( \text{alist.sort()} \) OR \( \text{result} = \text{sorted(alist)} \)
  
  - Now \text{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 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

<table>
<thead>
<tr>
<th>Sorted, won’t move</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>final position</td>
<td></td>
</tr>
</tbody>
</table>

Code for Selection Sort

def selectsort(data):
    for i in range(len(data)):
        minindex = minindex(i)
        data[i], data[minindex] = data[minindex], data[i]