Announcements

- No RQs until after Exam 2
- Assignment 6 due Thursday
- APT Quiz 2 Sun-Tue
- APT 7 due today, APT 8 out
- Exam 2 is April 11
- Lab this week!

Today:
- More practice with Dictionaries, finish last time

Lab this week - .csv files
- Answering questions about songs and movies

Rock n Roll America's Top 1,000 Classic Rock Songs
(Our Base Song List)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Song</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stairway to Heaven</td>
<td>Led Zeppelin</td>
</tr>
<tr>
<td>2</td>
<td>Hey Jude</td>
<td>Beatles</td>
</tr>
<tr>
<td>3</td>
<td>All Along the Watchtower</td>
<td>Hendrix, Jimi</td>
</tr>
<tr>
<td>4</td>
<td>Satisfaction</td>
<td>Rolling Stones</td>
</tr>
<tr>
<td>5</td>
<td>Like a Rolling Stone</td>
<td>Dylan, Bob</td>
</tr>
<tr>
<td>6</td>
<td>Another Brick in the Wall</td>
<td>Pink Floyd</td>
</tr>
<tr>
<td>7</td>
<td>Won't Get Fooled Again</td>
<td>Who</td>
</tr>
<tr>
<td>8</td>
<td>Hotel California</td>
<td>Eagles</td>
</tr>
<tr>
<td>9</td>
<td>Layla, Derek and the Dominos</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Sweet Home Alabama</td>
<td>Lynyrd Skynyrd</td>
</tr>
<tr>
<td>11</td>
<td>Bohemian Rhapsody</td>
<td>Queen</td>
</tr>
<tr>
<td>12</td>
<td>Riders on the Storm, Doors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rock and Roll, Led Zeppelin</td>
<td></td>
</tr>
</tbody>
</table>

Lab - .csv file
Python shortcut you can ignore

- The `zip` function, tuples from two lists
- Does something right if lists have different sizes. Look it up

```python
words = ['dog', 'cat', 'fish', 'guava']
counts = [3, 2, 1, 5]
cc = zip(words, counts)
[('dog', 3), ('cat', 2), ('fish', 1), ('guava', 5)]
```

Python functions you CANNOT ignore

- We know how to sort, we call `sorted`
  - Example: sorting tuples
  - Function `sorted` returns a new list, original not changed

```python
xx = [('dog', 3), ('cat', 2), ('fish', 1), ('guava', 2)]
yy = sorted(xx)
[('cat', 2), ('dog', 3), ('fish', 1), ('guava', 2)]
```

- What if sort by numbers instead of words?

```python
xx = [('dog', 3), ('cat', 2), ('fish', 1), ('guava', 2)]
```

Use what you know

- You can re-organize data to sort it as you'd like, list comprehensions are your friend

```python
xx = [('dog', 3), ('cat', 2), ('fish', 1), ('guava', 2)]
...nlist = [(t[1], t[0]) for t in xx]
[(3, 'dog'), (2, 'cat'), (1, 'fish'), (2, 'guava')]
yy = sorted(nlist)
[(1, 'fish'), (2, 'cat'), (2, 'guava'), (3, 'dog')]
```

APT – SortedFreqs

`bit.ly/101s17-0328-1`

The returned frequencies represent an alphabetic/lexicographic ordering of the unique words, so the first frequency is how many times the alphabetically first word occurs and the last frequency is the number of times the alphabetically last word occurs.
Ways to count?

• Dictionaries are faster than using lists?
• How fast is list.count(x) for each x?

Shafi Goldwasser

• 2012 Turing Award Winner
• RCS professor of computer science at MIT
  – Twice Godel Prize winner
  – Grace Murray Hopper Award
  – National Academy
  – Co-inventor of zero-knowledge proof protocols

How do you convince someone that you know [a secret] without revealing the knowledge?
• Honesty and Privacy

Work on what you like, what feels right, I know of no other way to end up doing creative work

Review Dictionaries

• Map keys to values
  – Counting: count how many times a key appears
    • Key to number
  – Store associated values
    • Key to list or set
• Get all
  – Keys, values or (key,value) pairs
• What question do you want to answer?
  – How to organize data to answer the question

Problem Statement

You will be given a string list names, containing a list of customer names extracted from a database. Your task is to report the customers that occur more than once in this list, and the number of occurrences for each of the repeated customers.


Returns: ["A 4", "B 2", "D 3"]
Dictionary problems
Number of students in ACM clubs
bit.ly/101s17-0328-4

d = {'duke':30, 'unc':50, 'ncsu':40}

d['duke'] = 80
d.update({'ecu':40, 'uncc':70})
print d.values()