Announcements

• No RQs until after Exam 2
• Assignment 6 due Thursday
• APT Quiz 2 due tonight
• APT 8 due today, APT 9 out
• Exam 2 is April 7
• Lab this week!

• Today:
  – More practice with Dictionaries
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

\[
\begin{align*}
xx &= [(\text{'dog'}, 3), (\text{'cat'}, 2), (\text{'fish'}, 1), (\text{'guava'}, 2)] \\
yy &= \text{sorted}(xx)
\end{align*}
\]

\[
[(\text{'cat'}, 2), (\text{'dog'}, 3), (\text{'fish'}, 1), (\text{'guava'}, 2)]
\]

– What if sort by numbers instead of words?
First use what you know

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

```python
x = [('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')]```
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.

```python
filename: SortedFreqs.py
def freqs(data):
    '''
    return list of int values corresponding to frequencies of strings in data, a list of strings
    '''

data = ['apple', 'pear', 'cherry', 'apple', 'cherry', 'pear', 'apple', 'banana']
Returns: [3, 1, 2, 2]
Ways to count?

1. How long does list.count(x) if you do this for each x?
2. How long does it take to compute a dictionary of counts
Shafi Goldwasser

- 2012 Turing Award Winner
- RCS professor of computer science at MIT
  - Twice Gödel 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**
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.

```python

Returns: ["A 4", "B 2", "D 3"]
```
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
d = {'duke':30, 'unc':50, 'ncsu':40}

d['duke'] = 80
d.update({'ecu':40, 'uncc':70})
print d.values()
Dictionary problems – part 2

• Consider the Python dictionary below on schools that map schools to number of students

\[
d = \{\text{'duke'}: 30, \text{'unc'}: 50, \text{'ncsu'}: 40, \text{'wfu'}: 50, \\
\text{'ecu'}: 80, \text{'meridith'}: 30, \text{'clemson'}: 80, \\
\text{'gatech'}: 50, \text{'uva'}: 120, \text{'vtech'}: 110\}
\]