CompSci 101
Introduction to Computer Science

April 6, 2017
Prof. Rodger
Lecture by Bo Li

Announcements

• Exam 2 Tuesday
• Reading and RQ start again after exam
• Assignment 7 due tonight, Assignment 8 out soon
• No Lab next week!
• No Consulting next Tuesday night
• Review Session: Mon. 7:15pm LSRC B101

• Today:
  – Finish slides from last time
  – Reviewing for the exam

Old Duke concert
Assignment 7 can be turned in Friday midnight with no penalty

About Me

• Bo Shi
• Master of Science
• Artificial Intelligence

• Light consulting hours tonight
• Extra hours on Friday afternoon posted on Piazza
import csv, operator

f = open('top1000.csv','rbU')
data = {}
for d in csv.reader(f,delimiter=',',quotechar='"'):
    artist = d[2]
song = d[1]
if not artist in data:
data[artist] = 0
data[artist] += 1

itemlist = data.items()

x = sorted(([t[1],t[0]) for t in dict.items()])
x = [(t[1],t[0]) for t in x]
x = sorted(dict.items(),key=operator.itemgetter(1))

print x[:30]
Two-pass (or more) sorting

- Because sort is stable sort first on tie-breaker, then that order is fixed since stable

\[
\begin{align*}
a_0 &= \text{sorted(data, key=operator.itemgetter(0))} \\
a_1 &= \text{sorted(a0, key=operator.itemgetter(2))} \\
a_2 &= \text{sorted(a1, key=operator.itemgetter(1))}
\end{align*}
\]

\[
\begin{align*}
a_0 &= [(\text{'f'}, 2, 0), (\text{'c'}, 2, 5), (\text{'b'}, 3, 0), \\
&\quad (\text{'e'}, 1, 4), (\text{'a'}, 2, 0), (\text{'d'}, 2, 4)] \\
a_1 &= [(\text{'a'}, 2, 0), (\text{'f'}, 2, 0), \\
&\quad (\text{'b'}, 3, 0), (\text{'d'}, 2, 4), (\text{'e'}, 1, 4)] \\
a_2 &= [(\text{'e'}, 1, 4), (\text{'a'}, 2, 0), (\text{'f'}, 2, 0), \\
&\quad (\text{'d'}, 2, 4), (\text{'c'}, 2, 5), (\text{'b'}, 3, 0)]
\end{align*}
\]

How to import: in general and sorting

- We can write: import operator
  - Then use key=operator.itemgetter(…)

- We can write: from operator import itemgetter
  - Then use key=itemgetter(…)

- Note: itemgetter is not on exam2, but will be on the final exam

Exam logistics

- Only need a pen or pencil
- No scratch paper
- See the reference sheet of Python information you will get with the test (see resources page)
- Closed book, closed notes, closed neighbor
- Covers lecture, lab and assigned reading
- Have put old RQ quizzes back up as quiz review
  - This is NOT for a grade, for studying only
Understand old and new topics

- Old topics: if, for, while, lists, strings
- list comprehension, enumerate
- Files – write code - Will give you a file already opened and ready for reading
- Sets, Dictionaries – write code – create and use
- Understand items on Python review sheet on resources page
- HAVE NOT COVERED TOPICS – regular expressions or recursion

The best way to study

- Write code on paper!
- Resources page has old tests and solutions – Try writing code, then look at solutions
- Rewrite an APT
- Rewrite code we did in lecture
- Rewrite code we did in classwork or lab

Looping by index or by element

- Strings and lists: use either
  - range(len(x)) for index, can get element
  - enumerate(somelist)
- Sets and Dictionaries: element only
  - Loop over d or d.keys() for dictionary
  - The keys are a set, so similar to set loop
- Which is best when choice? It depends!
  - Can you get element from index?
  - Can you get index from element?

Questions

bit.ly/101s17-0406-1
Unpacking a list comprehension

\[ f(x) \text{ for } x \text{ in } \text{foo if condition with } x \]
\[ w \text{ for } w \text{ in } \text{words if } w\text{.endswith('e')} \]
\[(w, \text{words.count}(w)) \text{ for } w \text{ in } \text{set(words)}\]

– Always possible to use a loop

```python
build = []
for x in foo:
    if condition with x:
        build.append(f(x))
build = []
for w in set(words):
    build.append((w, words.count(w)))
```

Set Concepts

• Set union, intersection, difference
  – s.intersection(t) is the same as s&t
  – s.union(t) is the same as s∪t
  – s.difference(t) is the same as s-t

• Sets aren't in order during iteration
  – Convert to list, create from list
  – Sets are really, really efficient for add/search

Dictionaries

• Build a dictionary
  – Counting dictionary
    • string to number
  – Grouping dictionary
    • string to list of items related

• Use a dictionary
  – Get values
  – Get keys
  – Get key,value pair

Questions

bit.ly/101f16s17-0406-2
Now go over Test Practice problems