CompSci 101
Introduction to Computer Science

April 6, 2017

Prof. Rodger

Lecture by Bo Li

Review for exam
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

Lil Jon to headline this year's Old Duke concert

By Claire Ballentine | Wednesday, March 29

Light consulting hours tonight

Extra hours on Friday Afternoon posted on Piazza
About Me

• Bo Shi

• Master of Science

• Artificial Intelligence
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()
dds = sorted(itemlist,key=operator.itemgetter(1),reverse=True)
print dds[:30]
Understanding sorting API

• How API works for `sorted()` or `.sort()`
  – Alternative to changing order in tuples and then changing back

```python
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))
```

• Sorted argument is key to be sorted on, specify which element of tuple. Must import library operator for this
Sorting from an API/Client perspective

• API is Application Programming Interface, what is this for sorted(..) and .sort() in Python?
  – Sorting algorithm is efficient, stable: part of API?
  – sorted returns a list, doesn't change argument
  – sorted(list, reverse=True), part of API
  – foo.sort() modifies foo, same algorithm, API

• How can you change how sorting works?
  – Change order in tuples being sorted,
    • [(t[1],t[0]) for t in ...]
  – Alternatively: key=operator.itemgetter(1)
Beyond the API, how do you sort?

• Beyond the API, how do you sort in practice?
  – Leveraging the stable part of API specification?
  – If you want to sort by number first, largest first, breaking ties alphabetically, how can you do that?

• Idiom:
  – Sort by two criteria: use a two-pass sort, first is secondary criteria (e.g., break ties)

[ ("ant", 5), ("bat", 4), ("cat", 5), ("dog", 4) ]
[ ("ant", 5), ("cat", 5), ("bat", 4), ("dog", 4) ]
Two-pass (or more) sorting

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

```python
a0 = sorted(data, key=operator.itemgetter(0))
a1 = sorted(a0, key=operator.itemgetter(2))
a2 = sorted(a1, key=operator.itemgetter(1))
data
[('f', 2, 0), ('c', 2, 5), ('b', 3, 0),
 ('e', 1, 4), ('a', 2, 0), ('d', 2, 4)]
a0
[('a', 2, 0), ('b', 3, 0), ('c', 2, 5),
 ('d', 2, 4), ('e', 1, 4), ('f', 2, 0)]
```
Two-pass (or more) sorting

```python
a0 = sorted(data, key=operator.itemgetter(0))
a1 = sorted(a0, key=operator.itemgetter(2))
a2 = sorted(a1, key=operator.itemgetter(1))
a0
[('a', 2, 0), ('b', 3, 0), ('c', 2, 5), ('d', 2, 4), ('e', 1, 4), ('f', 2, 0)]
a1
[('a', 2, 0), ('b', 3, 0), ('f', 2, 0), ('d', 2, 4), ('e', 1, 4), ('c', 2, 5)]
a2
[('e', 1, 4), ('a', 2, 0), ('f', 2, 0), ('d', 2, 4), ('c', 2, 5), ('b', 3, 0)]
```
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) for x in foo if condition with x]
[w for w in words if w.endswith('e')]
[(w, words.count(w)) for w in set(words)]

– Always possible to use a loop

```python
build = []
for x in foo:
    if condition with x:
        build.append(f(x))
```
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