Compsci 101
Sorting, CSV

Susan Rodger
November 10, 2022

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Rank</td>
</tr>
<tr>
<td>2</td>
<td>1 Like a Rolling Stone</td>
<td>Song</td>
</tr>
<tr>
<td>3</td>
<td>2 Satisfaction</td>
<td>Artist</td>
</tr>
<tr>
<td>4</td>
<td>3 Imagine</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4 What's Going On</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5 Respect</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6 Good Vibrations</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7 Johnny B. Goode</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8 Hey Jude</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9 Smells Like Teen Spirit</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>10 What'd I Say</td>
<td></td>
</tr>
</tbody>
</table>
S is for ...

• **Software**
  • Joy, sorrow, fun, changing the world

• **System and sys**
  • Connecting to the machine at different levels

• **Sorting**
  • From hat to tim to more
Barbara Liskov

• Among first women in US to earn Ph.D. in Computer Science: 1968
• Turing Award 2008, Software Engineering and Programming Languages
• Object-Oriented
  • CLU
• Liskov Substitution Principle

“Every time you exchange e-mail with a friend, check your bank statement online, or run a Google search, you are riding the momentum of her research” – MIT President Rafael Reif about Liskov
Announcements

• APT 5 due today!

• Assignment 5 due Thurs, Nov 17

• No Lab this week

• Reading and Sakai Quizzes due next week

• APT Quiz 2 – today through Monday
APT Quiz 2 Nov 10-14

• Opens 10/14 Noon
• Closes at 11pm 11/14 – must finish all by this time
• There are two parts based on APTs 1-5
  • Each part has two APT problems
  • Each part is 3 hours – more if you get accommodations
  • Each part starts in Sakai under tests and quizzes
  • Sakai is a starting point with countdown timer that sends you to a new apt page just for each part
  • Could do each part on different day or same days
• Old APT Quiz so you can practice (not for credit) – on APT Page
APT Quiz 2

• Is your own work!
  • No collaboration with others!
  • Use your notes, lecture notes, your code, textbook
  • DO NOT search for answers!
  • Do not talk to others about the quiz until grades are posted

• Post private questions on Ed Discussion
  • We are not online between 9pm and 9am!
  • We are not on all the time, especially weekends
  • Will try to answer questions between 9am – 9pm
    • About typos, cannot help you in solving APTs

• See 101 APT page for tips on debugging APTs
APT Quiz

Start the APT quiz on Sakai under quizzes, but not until you are ready to take the quiz.

APTs

See below for hints on what to do if your APT doesn’t run.

For each problem in an APT set, complete these steps by the due date:

- first click on the APT set below to go to the APT page.
- write the code, upload the file and click the Submit link
- check your grade on the grade code page by clicking on check submissions

In solving APTs, your program should work for all cases, not just the test cases we provide. We may test your program on your homework.

<table>
<thead>
<tr>
<th>APT</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>APT-1</td>
<td>Sept. 15</td>
</tr>
<tr>
<td>APT-2</td>
<td>Sept. 29</td>
</tr>
<tr>
<td>APT-3</td>
<td>Oct. 13</td>
</tr>
<tr>
<td>PRACTICE FOR APT QUIZ 1</td>
<td>PRACTICE ONLY</td>
</tr>
</tbody>
</table>

We may do some APTs partially in class or lab, but you still have to do them and submit them. There will usually be extra points. You can do more than required to challenge yourself. We do notice if you do more APTs than those required. If you do extra ones, they still have to be turned in on the due date.

Regrades

If you have concerns about an item that was graded (lab, apt or assignment), you have one week after the grade is posted to fill out a regrade form here.

Problems Running an APT? Some Tips!

Stuck! Use 7 steps!
Don't go to Sakai to start APT Quiz until you are ready to start

If you click on it, you start it!
PFTD

• Sorting
  • Sorting using standard Python APIs

• CSV Library
  • How to read data using standard Python APIs

• Lambda
  • Language construct to make sorting simpler (next week)
Song: Total Eclipse of the Heart, Bonnie Tyler
https://www.youtube.com/watch?v=lcOxhH8N3Bo
Why Sort Data?

- Help understand data
  - Great American Eclipse, August 21, 2017
  - Spotify tracked the playing of the song
Why Sort Data?

• Help understand data
  • Great American Eclipse, August 21, 2017
  • Spotify tracked the playing of the song
Why Sort Data?

- Every field needs to visualize and understand data.
- Sorting helps with this from movies to policy to sports to location of infections to


How your GIS department can respond to COVID-19

Local Government
March 09, 2020

Mike Schoelen

A staggering wealth of geospatial information has emerged regarding the COVID-19 outbreak. Dashboards, near real-time services, and GitHub repositories have built the foundation for an extraordinarily transparent response effort.
How To Sort: Algorithms

• Does scale matter? It depends!

• You're playing Spades, Hearts, Bridge, Go-Fish
  • How you sort doesn't really matter, but whether you sort makes play more efficient? Better?

• Many ways to sort
  • Bubblesort, Insertion sort, Selection sort
  • Quicksort, Mergesort, Timsort, Bogo sort
  • Python uses Timsort
WOTO-1 Popular Music

• Make a copy of this spreadsheet:

• Who are top two artists? Most Songs

• How did you do it?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rank</td>
<td>Song</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Like a Rolling Stone</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Satisfaction</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Imagine</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>What's Going On</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>Respect</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>Good Vibrations</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>Johnny B. Goode</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>Hey Jude</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>Smells Like Teen Spirit</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>What'd I Say</td>
</tr>
</tbody>
</table>
# WOTO-1 Popular Music


<table>
<thead>
<tr>
<th>Rank</th>
<th>Song</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Like a Rolling Stone</td>
<td>Bob Dylan</td>
</tr>
<tr>
<td>3</td>
<td>Satisfaction</td>
<td>The Rolling Stones</td>
</tr>
<tr>
<td>4</td>
<td>Imagine</td>
<td>John Lennon</td>
</tr>
<tr>
<td>5</td>
<td>What's Going On</td>
<td>Marvin Gaye</td>
</tr>
<tr>
<td>6</td>
<td>Respect</td>
<td>Aretha Franklin</td>
</tr>
<tr>
<td>7</td>
<td>Good Vibrations</td>
<td>The Beach Boys</td>
</tr>
<tr>
<td>8</td>
<td>Johnny B. Goode</td>
<td>Chuck Berry</td>
</tr>
<tr>
<td>9</td>
<td>Hey Jude</td>
<td>The Beatles</td>
</tr>
<tr>
<td>10</td>
<td>Smells Like Teen Spirit</td>
<td>Nirvana</td>
</tr>
<tr>
<td>11</td>
<td>What'd I Say</td>
<td>Ray Charles</td>
</tr>
</tbody>
</table>
Solve a Larger Problem

• Suppose I were to give you the top 1000 artists
  • Top 1,000 songs, find top 10 artists
  • How many songs per artist?
Scale

• **As the size of the problem grows we want ...**
  • The algorithm to still work and be fast!
  • What to do?

• **Search example**
  • Google search results work
  • SoundHound/Shazam results work
  • ContentID on YouTube results work
Python to the Rescue

• **Using `.sort(...)`, `sorted(...)`, and `lambda`**
• **Using CSV library and its API**
  • CSV – Comma Separated Values
• **Why use the CSV library?**
  • How to handle the song “Hello, I Love You”?
  • Row 166 in spreadsheet
Hits by Artists: SongReader.py

• What is returned by this function?
• details of csv: `next` and no `split` and ...

```python
def countByArtist(name):
    csvf = open(name, 'r', encoding='utf-8')
    freader = csv.reader(csvf)
    header = next(freader)
    print("header row labels", header)
    data = {}
    for row in freader:
        artist = row[2]
        if artist not in data:
            data[artist] = 0
            data[artist] += 1
    csvf.close()
    return data
```
Hits by Artists: SongReader.py

• What is returned by this function?
  • details of csv: `next` and no `split` and ...
WOTO-2 countByArtist
Two APIs: CSV and Sorting

- **CSV Library to read and process data**
  - Comma-separated, but can by ":" separated, or any character as we'll see later

- **Similar to reading a file – returned by open**
  - Iterable is returned by `csv.reader`
  - The `next` function advances iterable
  - Don't call `split`, we can access by index
    - Also by header-row label with `csv.DictReader`
CSV API

- `freader = csv.reader(file)` – returns an iterable
  - Every line from the file in a form ready for you
- `line = next(freader)`
  - Gives you next row as list of strings
- `for row in freader:`
  - Gives you the rest of rows as list of strings
What does this do? freader an iterable Where name is a filename

csvf = open(name, 'r', encoding='utf-8')
freader = csv.reader(csvf)
print("freader", freader)
header = next(freader)
print("header", header)
for row in freader:
    print("row", row)
What does this do? `freader` an iterable
Where name is a filename

csvf = open(name, 'r', encoding='utf-8')
freader = csv.reader(csvf)
print("freader", freader)
header = next(freader)
print("header", header)
for row in freader:
    print("row", row)

freader <csv.reader object at 034>
header ['Rank', 'Song', 'Artist']
row ['1', 'Stairway to Heaven', 'Led Zeppelin']
row ['2', 'Hey Jude', 'Beatles']
row ['3', 'All along the Watchtower', 'Hendrix, Jimi']
row ['4', 'Satisfaction', 'Rolling Stones']
...

11/10/22  Compsci 101, Fall 2022  26
What if you call \texttt{next} one extra time? Where \texttt{name} is a filename

\begin{minted}{python}
csvf = open(name, 'r', encoding='utf-8')
freader = csv.reader(csvf)
\texttt{print}("freader", freader)
header = \texttt{next}(freader)
\texttt{print}("header", header)
nextline = \texttt{next}(freader)
\texttt{print}("next", nextline)
for row in freader:
    \texttt{print}("row", row)
\end{minted}
What if you call `next` one extra time?
Where name is a filename

csvf = open(name, 'r', encoding='utf-8')
freader = csv.reader(csvf)
print("freader", freader)
header = next(freader)
print("header", header)
nextline = next(freader)
print("next", nextline)
for row in freader:
    print("row", row)

freader <csv.reader object at 034>
header ['Rank', 'Song', 'Artist']
nextline ['1', 'Stairway to Heaven', 'Led Zeppelin']
nextline ['2', 'Hey Jude', 'Beatles']
nextline ['3', 'All along the Watchtower', 'Hendrix, Jimi']
nextline ['4', 'Satisfaction', 'Rolling Stones']
...

Sorting to Print/Visualize

• Dictionary is ('Beatles', 51) tuples
  • But tuples not in order, so we must ...

```python
if __name__ == '__main__':
    counts = countByArtist("data/top1000.csv")

    print('First 5 artists: ')
    for artist in sorted(counts.items())[:5]:
        print(artist)

    print('Top 5 artists: ')
    sortbycount = sorted([(a[1], a[0]) for a in counts.items()])
    sortedArtists = [(a[1], a[0]) for a in sortbycount]
    for artist in sortedArtists[-5:]:
        print(artist)
```
Sorting to Print/Visualize

• Dictionary is ('Beatles', 51) tuples
  • But tuples not in order, so we must ...

```python
if __name__ == '__main__':
    counts = countByArtist("data/top1000.csv")
    print('First 5 artists: ')
    for artist in sorted(counts.items())[:5]:
        print(artist)

    print('Top 5 artists: ')
    sortbycount = sorted([(a[1], a[0]) for a in counts.items()])
    sortedArtists = [(a[1], a[0]) for a in sortbycount]
    for artist in sortedArtists[-5:]:
        print(artist)
```

What is going on here?

Why more complicated than lines 28 & 29?
WOTO-3 Calling countByArtist
Sorting API and Sorting Concepts

• **What is `counts.items()` – how is it sorted?**

```python
print('
First 5 artists:
for artist in sorted(counts.items())[:5]:
    print(artist)
```

• **What does sorted return?**
  • A list, you can slice a list, look for clues!
  • What can be sorted? A sequence
  • `sorted(counts.items())`
Sorting API and Sorting Concepts

• **What is counts.items() – how is it sorted?**

```python
print('\nFirst 5 artists: ')
for artist in sorted(counts.items())[0:5]:
    print(artist)
```

• **What does sorted return?**
  • A list, you can slice a list, look for clues!
  • What can be sorted? A sequence
  • `sorted(counts.items())`
Sorting by Number of Songs

• Sort by first value vs sort by second value
  • Need to put sequence back to original format

```python
print('\nFirst 5 artists:')
for artist in sorted(counts.items())[:5]:
    print(artist)

print('\nTop 5 artists:')
sortedArtists = sorted([(a[1], a[0]) for a in counts.items()])
for artist in sortedArtists[-5:]:
    print(artist)
```
Sorting by Number of Songs

- Sort by first value vs sort by second value
- Need to put sequence back to original format

```python
print('\nFirst 5 artists:"
for artist in sorted(counts.items())[:5]:
    print(artist)

print('\nTop 5 artists:"
sortedArtists = sorted([[a[1], a[0]] for a in counts.items()])
sortedArtists = [(a[1], a[0]) for a in sortedArtists]
for artist in sortedArtists[-5:]:
    print(artist)
```

If we comment out 33, what's printed? Why?
Python Sorting API

• We'll use both `sorted()` and `.sort()` API
  • How to call, what options are
  • How to sort on several criteria

• Creating a new list, modifying existing list
  • `sorted(..)` creates list from .. Iterable
  • `x.sort()` modifies the list x, no return value!