Compsci 101
Sorting, CSV
Live Lecture

Susan Rodger
Nicki Washington
March 25, 2021

Announcements
• Nothing due today!
• Assignment 4 due Tue, March 30
• APT-6 due Thur, Apr 1
• Assignment 5 out today, due Tues, Apr 6
  • Discuss on March 30th
  • Read it before then, also a Sakai quiz out today to go with it
• Lab 8 on Friday
  • There is no prelab!

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)

WOTO-1 Popular Music
• Make a copy of this spreadsheet:
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>Rank</th>
<th>Song Name</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Like a Rolling Stone</td>
<td>Bob Dylan</td>
</tr>
<tr>
<td>2</td>
<td>Satisfaction</td>
<td>The Rolling Stones</td>
</tr>
<tr>
<td>3</td>
<td>Imagine</td>
<td>John Lennon</td>
</tr>
<tr>
<td>4</td>
<td>What's Going On</td>
<td>Marvin Gaye</td>
</tr>
<tr>
<td>5</td>
<td>Respect</td>
<td>Aretha Franklin</td>
</tr>
<tr>
<td>6</td>
<td>Good Vibrations</td>
<td>The Beach Boys</td>
</tr>
<tr>
<td>7</td>
<td>Johnny B. Goode</td>
<td>Chuck Berry</td>
</tr>
<tr>
<td>8</td>
<td>Hey Jude</td>
<td>The Beatles</td>
</tr>
<tr>
<td>9</td>
<td>Smells Like Teen Spirit</td>
<td>Nirvana</td>
</tr>
<tr>
<td>10</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')
    reader = csv.reader(csvf)
    header = next(reader)
    print("header row labels", header)
    data = {}
    for row in reader:
        artist = row[2]
        if artist not in data:
            data[artist] = 0
            data[artist] += 1
    csvf.close()
    return data
```

What is new? What does it do?

What is returned by this function?

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

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

What is going on here?

Why more complicated than lines 28 & 29?
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`

Sorting API and Sorting Concepts

- What is `counts.items()` – how is it sorted?
  - `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 by Number of Songs

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

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!
API to change sorting

- In SongReader.py we changed order of tuples to change sorting order
  - Then we sliced the end to get "top" songs

- Can supply a function to compare elements
  - Function return value used to sort, key=function
  - Change order: reverse=True

Sorting Examples

- Use key=function argument and reverse=True
  - What if we want to write our own function?

```python
In[2]: a = ['red', 'orange', 'green', 'blue', 'indigo', 'violet']
In[3]: sorted(a)
Out[3]: ['blue', 'green', 'indigo', 'orange', 'red', 'violet']
In[4]: sorted(a, key=len)
Out[4]: ['red', 'blue', 'green', 'orange', 'indigo', 'violet']
In[5]: sorted(a, key=len, reverse=True)
Out[5]: ['orange', 'indigo', 'violet', 'green', 'blue', 'red']
```

WOTO-4 Sorting

Assignment 5 – Clever Hangman

- Must finish Hangman assignment first!
- This is a copy and modify

- Make it hard to win
- Keep changing the word
  - More than that
- We will discuss next time