

Assignment 3 - The Hits from Coast to Coast

The goal of this assignment is to demonstrate your mastery of data structures by using a data structure as part of a larger software system. This assignment has one implementation requirement detailed below.

Background

The songs played on American radio stations get compiled weekly. These songs may be played on [American Top 40](#) and other radio shows and are also compiled on web sites. Examining the history of these, you may find that the most popular songs have lately been collaborations between pairs of musical artists. For example, the song “Levitating” by the singer Dua Lipa features the artist DaBaby. Dua Lipa also has a song “Cold Heart” with Elton John. We may conjecture that if a song is performed (or features) DaBaby, we may also like a song by Elton John because of their mutual connection to Dua Lipa. In this assignment, you will produce a number of artist recommendations based on collaborations of those artists with others.

Part 1 - Data

Use JSoup or your own code to download data from Top40 Weekly, available at <https://top40weekly.com/>. You may limit the time period to the 1990s, 2000s, 2010s and 2020s since collaborations before this period rarely made the top 40 charts.

JSoup is a Java library for working with real-world HTML files. You may download the JSoup JAR via <https://jsoup.org/> and integrate it into your IDE as described at [this Stack Overflow page](#) for IntelliJ or [this page](#) for Eclipse. There are many tutorials on JSoup. You may use it with code similar to the following:

```
import org.jsoup.nodes.Document;
import org.jsoup.Jsoup;
Document document = Jsoup.connect("http://top40weekly.com").get();
String page_content = document.text();
```

... which will download the page at the URL provided and render the page as plain text and persist it in the data structure called “document”. The page <https://www.baeldung.com/jsoup-line-breaks> may provide information on how to preserve line breaks, determine URLs, and provide other processing capabilities from a downloaded page. You should store the downloaded data in a data structure of your choice for the next part.

Each page on the top40weekly site contains the Top 40 songs for each week of the year, in order of popularity one per line. Figure 1 has two lines from one such page. Each line contains the following fields

(in order): the current week's chart position; the previous week's chart position (if it exists or a “_” if not); the name of the song, a separator (similar to “_ . _”) and the artist or artists. Many of these songs are repeated from one week to the next. In addition, there are additional songs which may or may not appear in the Top 40 — specifically, songs listed under “THIS WEEK’S DROP”, “POWER PLAYS” and “NEW THIS WEEK” or others. You may choose to ignore or include these.

```
1 1 THRIFT SHOP – • – Macklemore & Ryan Lewis featuring Wanz (label)-17 (1) (2 Weeks at #1)
2 2 LOCKED OUT OF HEAVEN – • – Bruno Mars (label)-17 (1) (6 Weeks at #1)
```

Figure 1: 2 entries in the Top 40 (Week Ending February 9, 2013)

Your implementation must read the data from these web pages and persist enough of the data to complete the remainder of this assignment.

Part 2 - Artist Recommendations

Your implementation must accept the name of a musical artist as input from the user. If the artist has not collaborated with others on any songs, your implementation must produce no recommendations. However, if the artist has collaborated with others, your implementation must produce a list of all the collaborators, all of collaborators' collaborators, etc. This output may be in any order. For example, if the user requests collaborators of the artist Cardi B, the recommendations must include:

- A\$AP Rocky
- G-Eazy
- Maroon 5
- Nicki Minaj
- Migos

... as well as collaborators with Migos:

- Lil Uzi Vert
- YFN Lucci
- Trouble

... collaborators of Nicki Minaj:

- Drake
- Jason Derulo
- Ty Dolla \$ign
- Ariana Grande
- Nick Jonas
- Major Lazer
- Gucci Mane
- Katy Perry
- Lil Wayne

... etc.

Submission

No starter code is provided. Check your Java code — including any main function, interfaces and class implementations — into a GitHub repository. Submit the link to this GitHub repository on Canvas.

Grading

Your grade for this assignment will be determined as follows:

- 60% = Implementation: your class implementations must run successfully with valid input arrays. The implementation must produce the expected results. Any deviation from the expected results results in 0 credit for implementation.
- 25% = Decomposition: in the eyes of the grader, your implementation must demonstrate a reasonable object oriented decomposition — i.e. encapsulation, polymorphism and inheritance — and the proper choice of data structure(s) for this problem.
- 5% = Efficiency: in the eyes of the grader, your implementation must be maximally efficient with respect to running time and required space.
- 10% = Style: in the eyes of the grader, your implementation must be well-commented, use intelligently-named variables and functions.

Get started early. Avoid submitting late. If you do not have an instructor/SDS-approved accommodation and submit late, you will use 1 of your Grace Days for each day late. Once Grace Days have been exhausted, the penalty for late submission is 25% per day.