

Project PlaylistApp

Create a popular playlist application that accepts song playlists along with their popularities, stores the most popular 1024 playlists, and i) suggests playlists given a song and ii) suggests songs when the user starts typing a song name.

The playlist application should support the following operations:

- *loading of at most 128 playlists along with their popularities as a text file*: the format of the text file will be, in each line `<Playlist \t Popularity(integer)>` where *Playlist* is a space separated list of song names and there are 128 or less lines in the file. Note that if the number of playlists exceeds 1024, your system should remove the least popular playlists to keep the size in 1024.
- *loading of a single playlist along with its popularity through a form page*: Your app will have a page/tab with a form that enables addition of a single playlist along with its popularity to your popular playlist store. Note that if the number of playlists exceeds 1024, your system should remove the least popular playlists to keep the size in 1024.
- *listing top 8 most popular playlists*: Display the most popular 8 playlists to the user.
- *given a song, suggest the most popular playlist that song is in*.
- *as the user starts typing a song-name, the system should start suggesting four song names based on the popularity of the songs. The popularity of a song is the sum of the popularities of the playlists it is listed in. Note that as new playlists are added, popularities of songs change as well*.

An example song auto-completion is as follows: As the user types "all", the system suggests "all of me", "all about the bass", "all along the watchtower", "all that remains" as these four are the most popular songs that start with the characters "all", but as the user types "all o" the suggestions are now "all of me", "all of me loves all of you", "all of my life", "all of the stars" as now these four are the most popular songs that start with the characters (have the prefix) "all o".

You are free to use any data structures you like for storing the songs, the playlists, etc... But you must define the complexity of each functionality in your system, e.g. loading the playlists, listing top playlists, retrieving playlists, auto-completing/suggesting songs, maintaining song/playlist popularities etc... Efficient data structure selection & usage will get higher points.

Project Milestones [Oct 21st, Dec 1st, Dec 7th]

- Develop a GUI that can load playlists both from a text file and through your UI, and then list the most popular 8 playlists. [Due: Oct 21st, send a link to a video you made presenting your app.]
 - Completed project that supports all desired operations. [Due: Dec 1st, again send a link to a video you made presenting your app and send a project report.].
 - Project presentations [Due: Dec 7th]
-

Comparative Features

Your algorithms and data structures will be compared against others in the class and ranked according to:

- Speed/efficiency of key operations