Similar Anime Recommender

Team Members: Elliot Liu, Maxwell Griffin, Efren Lopez

GitHub: https://github.com/EfrenL0pez/Similar-Anime-Recommender
Youtube: https://youtube.com/ourvideo

Team 17
Kapoor
COP 3530
August 2nd, 2022

Extended and Refined Proposal

Problem: What problem are we trying to solve?

The website MyAnimeList, which is one of the most popular anime databases, has a feature where users can mark certain anime as being similar to other anime. Currently, there is no way to efficiently filter similar anime or search through them.

Motivation: Why is this a problem?

People generally like to watch things that are similar to what they have already watched, and user-generated similarity data can be a powerful tool to generate anime recommendations. However, there is no automated tool that can isolate, rank, and present the highest rated similar anime to the end user, and it is very inconvenient to scroll through hundreds of anime that users marked as similar, most of which are mediocre.

Features implemented

A program that solves this problem should be able to take in the name of an anime as input, quickly retrieve similar titles, sort them based on factors such as popularity and average rating, and present them to the user along with relevant information such as a synopsis. The user should also have access to filters that can restrict returned titles to certain years, dates, and formats (movies, or shows).

Description of data

We are using a public dataset that is available on kaggle.com. Kaggle is a data science community that allows users to search and share data sets, build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

Information on the dataset

- Anime information for 13,379 animes
- Anime IDs are not evenly distributed
- User information for 1,123,284 myanimelist users
- 214,271 interactions between anime pairs (recommended and related animes)
- 5,048,994 interactions between user pairs (friendship)
- 223,812,614 interactions between users and animes
- The dataset is the largest anime dataset on Kaggle
- File type CSV
- Total file size is 2GB
- Created by SVANO

Lorem ipsum

Tools/Languages/APIs/Libraries used

Tools:

- Adobe Illustrator to create the final PDF document deliverable
- C++ as our choice of language to implement file I/O and data storage
- VS code, VS Studio, and CLion as our IDE's
- zoom to record our video
- YouTube to upload our video

Libraraies:

- Standard C++ Library

Algorithms implemented

We will be representing each anime as a class that would store all of its information, such as its title, rating, and airing date, and storing every anime object in either a hash table or a red black tree. We will compare the insertion and search speeds of a hash table and a red black tree. Search results could be stored in a trivial data structure such as an array and sorted with insertion sort due to their small size.

Distribution of Responsibility and Roles: Who did what?

Elliot is reponsible for the implentation of the hash table, Maxwell is responisble for the implementation the red-black tree, and Efren is responsible for the final project deliverables.

Analysis [Suggested 1.5 Pages]

Hash Table

The hash table insertion for both the main hash table and the dictionary is O(n) in the worst case where n is the number of anime in the dataset because the anime IDs could all hash to the same value, and the program would need to traverse a linked list with n elements to insert a single element because the hash tables use separate chaining. The search and find functions are both O(n) in the worst case, where n is the number of anime in the dataset. This is because the hash tables utilize separate chaining. Retrieving related anime is O(n^2) in the worst case, where n is equivalent to the number of related anime to the search term because std::sort is used to sort the anime by score. The function also has to look up the anime and retrieve its information, but this is at worst O(n), so the complexity of sorting the related entries dominates. The function ReadFiles is also O(n) because reading the dataset and inserting the data are both O(n) in the worst case. The hash functions used are both O(1) in the worst case because only constant operations are performed.

Red Black Tree

Reflection

As a group, how was the overall experience for the project?

Our experience went well, everyone in the team participated and contributed from the beginning to the end. The work was distributed equally and agreed upon by all members from the beginning to ensure everyone understood their role in the project. After teams were officially established, our team agreed to meet up in person after class or virtually one a week until the project was due. This meeting was specifically to work on the project, update team members on status, and address problems or concerns. This was the highlight of our team, we did not want the final project to interfere with any of our studies, so we created a timeline of what work needed to be done every week to ensure the project would be completed by 3 days prior the due date. Organizing our goals helped the team stay on track at an early stage.

Did you have any challenges? If so, describe.

One of the challenges the team faced at in early stage was deciding what problem were going to solve. It was an interesting moment for us since we did not know each other, we could not come to an agreement as to what problem the team was going to solve. Another challenge the team faced was not communicating enough during the design and implementation phase of our project. Initially, we did not have a set structure for all of us to follow during the design phase. This ultimately created multiple issues during the implementation phase where we had to combine all our functions with our unique parameters and syntax.

If you were to start once again as a group, any changes you would make to the project and/or workflow?

We learned how important communication is, we should have addressed the importance of communicating at an early stage. We spent more time than expected trying to implement each other's functions in the later stages.

Comment on what each of the members learned through this process.

Elliot

Improved his understanding on the hash table data structure and learned that if a dataset is highly rated by many users, does not mean it is good dataset.

Max

Strengthen his understanding on how to parse data form a csv file and learned to properly implement, insertion, and balancing of a red-black tree.

Efren

Improved his communication skills and time management skill and learned the fundementals of data manipulations.

References

[1]"Hash Functions", *Cse.yorku.ca*, 2022. [Online]. Available: http://www.cse.yorku.ca/~oz/hash.html. [Accessed: 24- Jul- 2022].

[2]"Reference - C++ Reference", *Cplusplus.com*, 2022. [Online]. Available: https://cplusplus.com/reference/. [Accessed: 18- Jul- 2022].

[3]"MyAnimeList Dataset", *Kaggle.com*, 2022. [Online]. Available: https://www.kaggle.com/datasets/svanoo/myanimelist-dataset?resource=download&select=anime_anime.csv. [Accessed: 10- Jul- 2022].

[4]"IEEE - Reference Guide", *Ieee-dataport.org*, 2018. [Online]. Available: https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf. [Accessed: 22- Jul- 2022].