**DSA FINAL PROJECT REPORT**



By

Jayson Mikael

Bernard Wijaya

Nathanael Setiawan

**Binus International University**

**Jakarta**

**Anime Search Algorithm**

Name : Jayson Mikael

NIM : 2440032442

Name : Bernard Wijaya

NIM : 2440032410

Name : Nathanael Setiawan

NIM : 2440042866

**Table of Content**

**I. Main Problem**…………………...……………………………….................................Page 1

• The main problem for this project

**II. Our Solution**…………………………………………………………………...……..Page 1

• The solution for the main problem

**III. Alternative Data Structures and Algorithms**…….…....................................Page 1

• Alternative data structures and algorithms to our solution

**IV. How to Execute Program**…...…………………………………………………….Page 1

• How to execute our program

**V. Program Results**…….……………………………………………………………….Page 5

• The result of the program running

**VI. Project Links**………….…………………………………………………………….Page 6

• Links of this project

**I. Main Problem**

With our anime database having more than 12,000 entries and growing, we wanted to test and see which searching algorithm was the best. The entries are unordered by themselves and so we wanted to compare algorithms to see which one is the best, and if we can create our own algorithm.

**II. Our Solution**

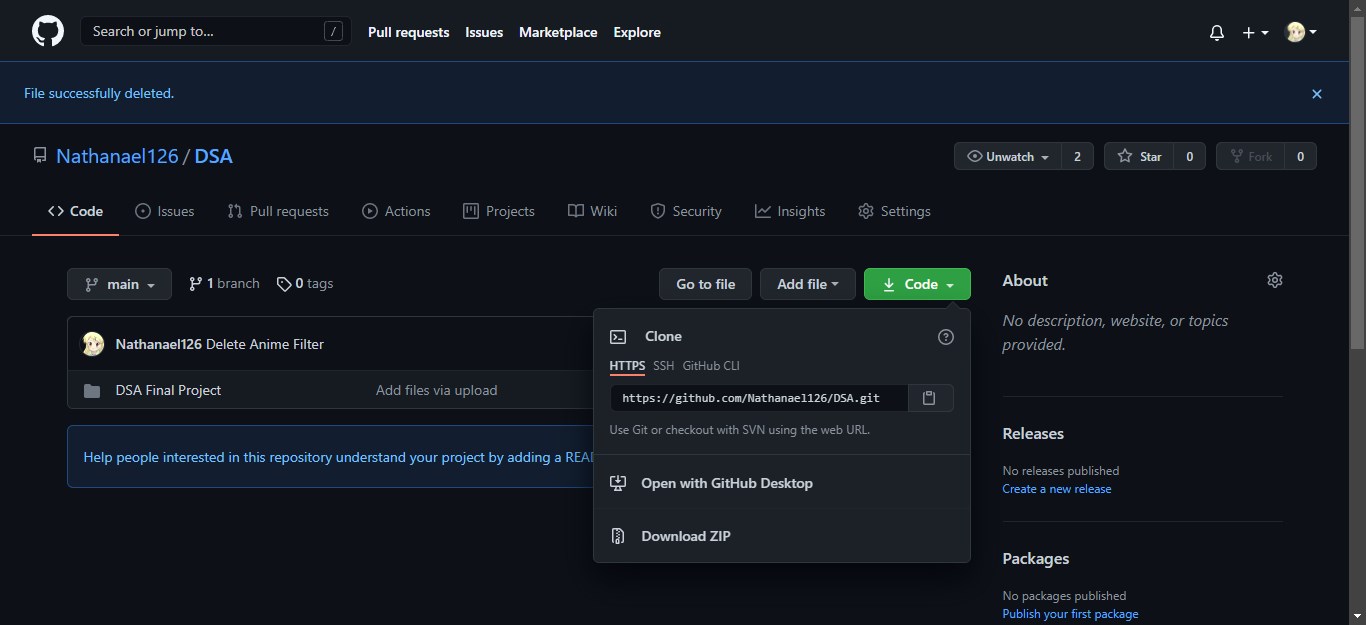
We decided to test a hashtable with our own hash function. This hash function takes the ascii values of each character in a given string, multiplies the values with its position index starting at 2, and adds these values to a total. We decided to compare this function to other searching algorithms, these algorithms are linear and binary search, and we compiled all these functions into 1 convenient program.

**III. Alternative Data Structures and Algorithms**

Linked list: A way to prevent collisions in hash tables. This is done by appending the data that would have collided with the other data at the end of a linked list. This is much more difficult when using the map data structure. Linked lists could have also helped to show more data without using separate data structures.

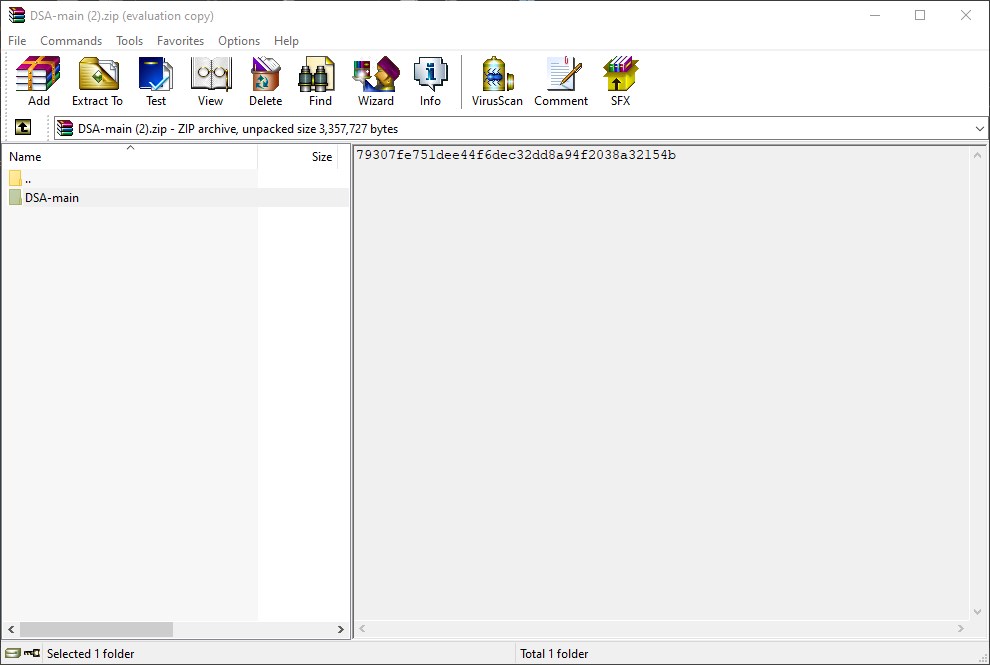
**IV. How to Execute Program**

*Step 1, Downloading from GitHub:*

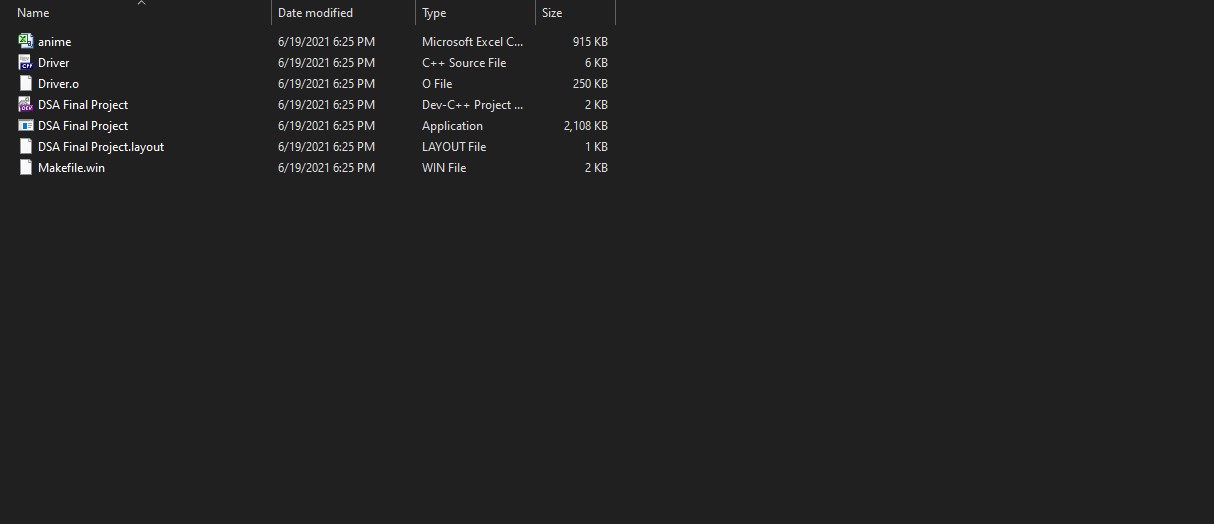
Use the link below to navigate towards the GitHub page, and download the code by first clicking the green button on the middle-right of the screen that says “Code” as well as having an arrow sign. It should open a dropdown menu, navigate to the bottom of the menu and click “Download ZIP”. It should start downloading a ZIP folder which can be seen in your browser. (Bottom of screen for Chrome)

*Step 2, Extract files:*

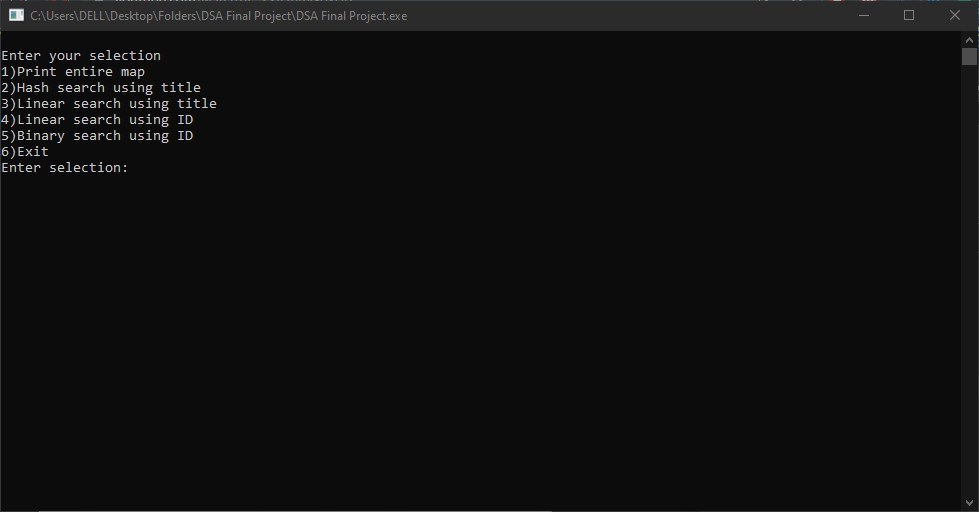
Using a program (such as WinRar), extract the folder contained inside the ZIP file into a directory. This can be done by dragging and dropping the folder onto another directory.



*Step 3, Launch program:*

Navigate through both folders in the previously selected directory by double-clicking both times on the folder. Locate the file titled “DSA Final Project.exe”, double click the program icon to launch the program.

The resulting screen should look like this (this screen will be referred to as the menu):



***Additional instructions:***

*Warnings:*

* Do NOT input a string into any functions that accept only integers. Doing so will cause an infinite loop to happen.
* There are still some bugs which we might have not identified yet, so please try to use the program as it is intended (using correct input types).

***How to search different things:***

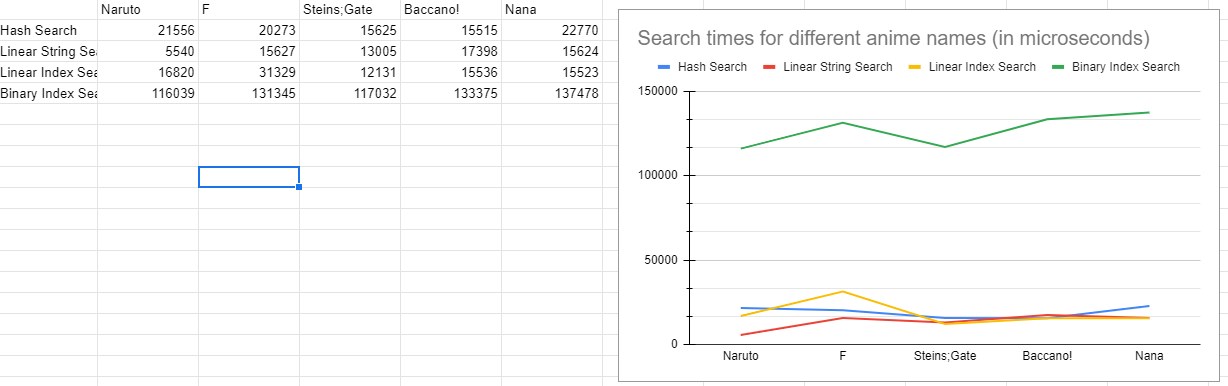
*Index search:*

1. In the menu, input “4” or “5” to search using an index (Integer).
2. Input an integer

*Title search:*

1. In the menu, input “2” or “3” to search using the title (String).
2. Input a string (This is case sensitive)

**V. Program Results**

****

**VI. Project Links**

Video demonstration: <https://drive.google.com/file/d/1FLrkqt1bI-BsXFnzyfsfGpg3RrCkfZFw/view?usp=drivesdk>

GitHub repository:

<https://github.com/Nathanael126/DSA>