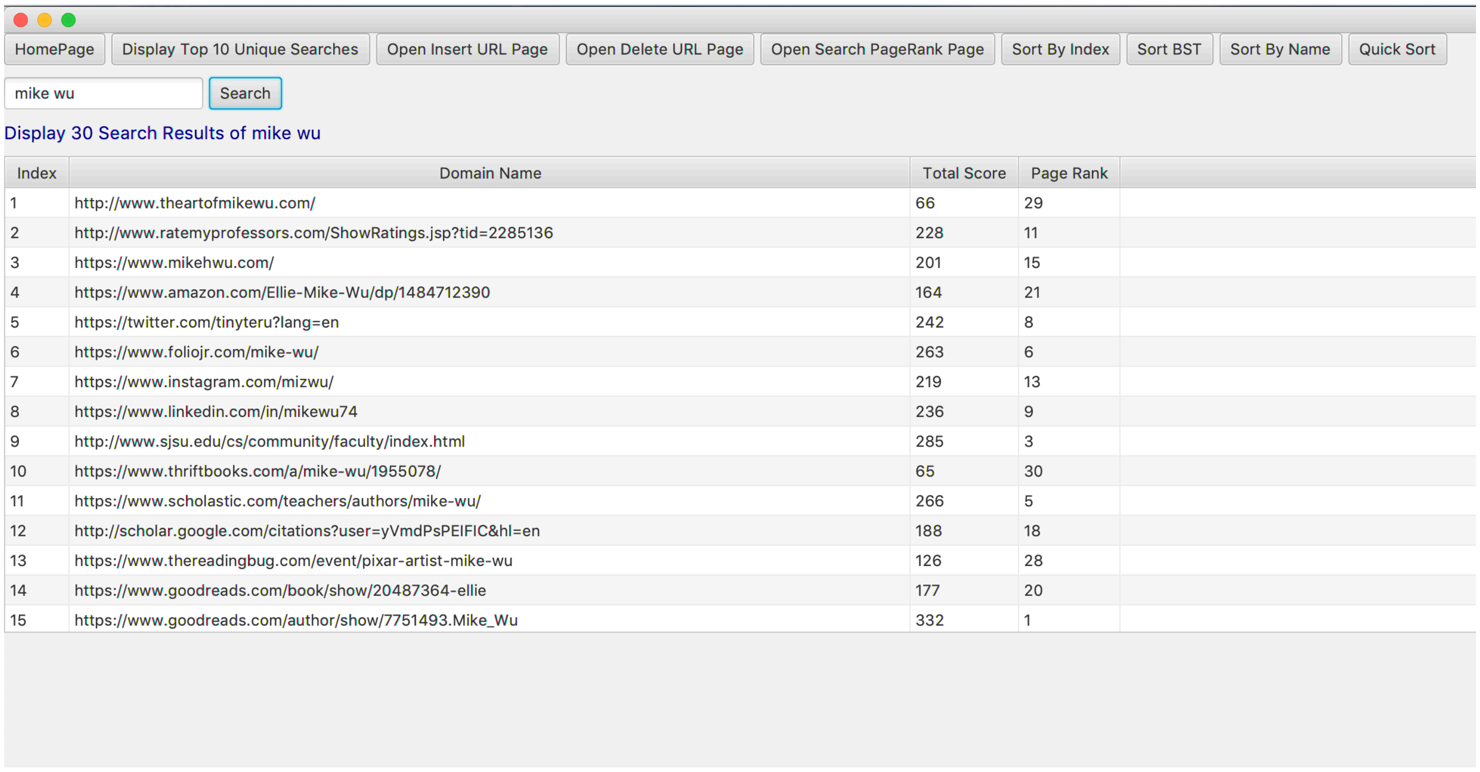
Programming Assignment 2

Google Search Engine Simulator

CS 146:S7 - Professor Mike Wu



Display 30 Search Results of mike wu

Index	Domain Name	Total Score	Page Rank
1	http://www.theartofmikewu.com/	66	29
2	http://www.ratemyprofessors.com/ShowRatings.jsp?tid=2285136	228	11
3	https://www.mikehwu.com/	201	15
4	https://www.amazon.com/Ellie-Mike-Wu/dp/1484712390	164	21
5	https://twitter.com/tinyteru?lang=en	242	8
6	https://www.foliojr.com/mike-wu/	263	6
7	https://www.instagram.com/mizwu/	219	13
8	https://www.linkedin.com/in/mikewu74	236	9
9	http://www.sjsu.edu/cs/community/faculty/index.html	285	3
10	https://www.thriftbooks.com/a/mike-wu/1955078/	65	30
11	https://www.scholastic.com/teachers/authors/mike-wu/	266	5
12	http://scholar.google.com/citations?user=yVmdPsPEIFIC&hl=en	188	18
13	https://www.thereadingbug.com/event/pixar-artist-mike-wu	126	28
14	https://www.goodreads.com/book/show/20487364-ellie	177	20
15	https://www.goodreads.com/author/show/7751493.Mike_Wu	332	1

Calvin Nguyen

Fall 2018

Table of Contents

Programming Assignment 2

Google Engine Search Simulator

Summary

Google's search engine is a powerful tool. Without search engines like Google, it would be practically impossible to find the information you need when you browse the Web. Like all search engines, Google uses sorting algorithm, indexing, searching, and priority queueing technique to generate search results. Google has a large index of keywords where those words can be found and uses automated programs called spiders or crawlers. What sets Google apart is how it ranks search results, which in turn determines the priority order Google displays results on its search engine results page (SERP). Google uses a trademarked algorithm called PageRank, which assigns each Web page a relevancy score.

Functional Requirement

The purpose of this project is to try to simulate Google Search Engine as much as I can by using different Data Structures and Sorting Algorithm. In this project, the main purposes are:

- Use Quicksort algorithm to sort the scores for PageRank and display in ascending order by PageRank
- Use Binary Search Tree Data Structure to manipulate the data
- Display Top 10 most searched keyword and sort the domain names of the current search keyword in ascending alphabet.
- The program is written in Java and the Data Structure needs to be followed the pseudocode from the textbook
- Use Web Crawler to get the real URL from Google API
- Every URL will have a total score, generated by 4 factors
- All the classes and functions will have a comment

Getting Started

Open a JAR File

1. Before download the zip file, make sure you have these things download to your computer:
    - IDE (Netbeans, Eclipse, IntelliJ). I highly recommend IntelliJ
    - Java SDK 8 & Java version 1.8
2. Unzip PA2-Cuong-Nguyen.zip
3. Open my folder, you can see PA2-Cuong-Nguyen.jar file.
4. Double click the .jar file to run the application.

Open Source Code From IDE

1. Open any IDE (Prefer IntelliJ) —> Open my project
2. Click File —> Project Structure —> Libraries —> Click '+'
3. Choose Java —> Choose "Lib" folder within my project
4. Click "Open" —> Click "OK"
5. Direct to src/Main.java to run the program.

Design & Implementation

In this section, I will explain how I built my application from scratch and why I decided to design like this. I created 6 classes because each class will only do its own functions. Hence, it will make more sense for me and for readers to easily understand the code and what each class does. For example:

- **Main** will only run the UI and run all functions whenever the button is clicked.
- **Sorting Algorithm** will be used to sort.
- **BSTree** will only have its unique functions
- **Link** will only store required data of the URL
- **Node** will store keyword and number of search of that keyword
- **Web Crawler** will do its scanning and generate BSTree for the search keyword

1. First and foremost, I built a **Link** to store the data of the URLs and **BSTree** class that will consist of 30 Links with its searchKeyword.
2. Then, I created **Web Crawler** class to make sure the web crawler is scanning and give me the URLs I want. Here, I decided to have a separate class for Web Crawler because I think that the web crawler will only have 1 function - call the Google API with the keyword and built a BSTree with 30 URLs. I don't want to put this Web Crawler inside the BSTree, because it will not make sense if BSTree can scan the keyword and call Google API
3. Next, after insert all the Links to the BSTree using WebCrawler, I begin to **sort the BSTree** by ascending Index, by ascending PageRank. Here, to know the pageRank of each Link, I need to finish import all 30 URLs first, and then sort the Tree based on the highest score. By doing that, I use a reversed version of in-order traverse. Instead of going from the left, I go from the right because when I insert the Link, I insert based on the score.
4. When I'm done with BST sort, I start working on QuickSort and BucketSort, so I created a separate class that will only do sorting.
    - **QuickSort**: I just use pseudocode and change compare by integer to compare the totalScore of the Links
    - **BucketSort:** in order to sort by domain name, I had to use the help of function getDomainName() on StackOverflow "**https://stackoverflow.com/questions/9607903/get-domain-name-from-given-url**" to get the DomainName of the URL. Then, I created a HashMap to store the Character and ArrayList. By doing this, I put all the domain that start with a corresponding character to the same bucket 'a', 'b', 'c', … 'z'. Then, I used Insertion sort to sort each bucket, and put it into the new ArrayList.
5. Now, the backend is done. I started to build the **UI** for the application. This is the hardest part because I had to pass the value from the backend to display in the Frontend.
    - I have **5 main Scenes**: HomePageScene, Display30ResultsScene, DisplayTop10SearchScene, insertURLScene, deleteURLScene, searchByPageRankScene. Each scene will allow the user to do different tasks. For instance, insertURLScene will allow

users to insert the URL and 4 factors of their choice, deleteScene will let the user pick the PageRank they want to delete.

- Then, in each scene, I will create **different buttons** and different text box depend on the purpose of the scene. For example, the homepage will only have a text box and a search button. Insert Page will have 5 text boxes (1 URL, 4 factors) and an insert URL button.
- Finally, in order to display the data. I used **TableView** to display the data.
- I created a **SearchList HashMap** to keep track if the keyword is already searched or not. If yes, we can just get the BSTree of the keyword in the HashMap without generate the Web Crawler again.

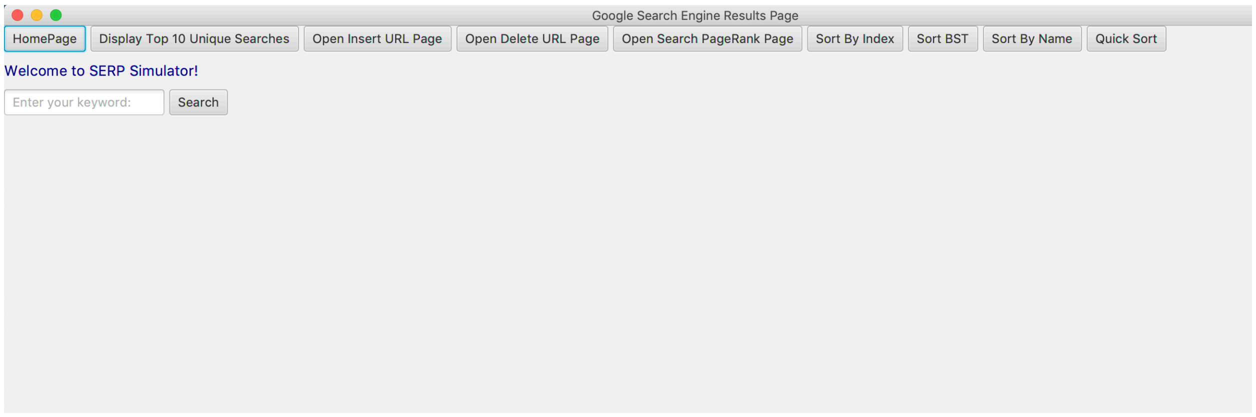
Classes, Subroutine, Functions

1. **Class "Link" will store all the required data of each URL**
    - Variables
        - Title: Title of the URL
        - URL: The url that is gotten from the Web Crawler
        - TotalScore: generated by 4 random numbers
        - Index: The position that the URL is inserted into the BSTree
        - PageRank: each URL will have a page rank based on its total score in the tree
        - Link parent, left, right: Each link will have its parent, right/left child because it is built in Tree Data Structure
    - Functions
        - Constructor, Get/Set Method of all private variable
        - generateRandomScore: generate 4 random integers and add all of them together
        - CompareTo: compare each Link based on the index
2. **Class "Node": store data of a keyword and the keyword's total searches**
    - Variables
        - SearchName: the keyword that user searches
        - totalSearch: The total number that people search for this keyword
    - Functions
        - Constructor, Get/Set Method of all private variable
        - CompareTo: compare each Node based on the totalSearch
3. **Class "WebCrawler": will read the keyword and fetch data of the keyword from Goole API**
    - Functions
        - createBST: will create a BSTree for the search keyword including all the URLs of that keyword into the BSTree
4. **Class "Sorting Algorithm": will contain Quicksort and BucketSort algorithm**
    - Functions
        - quickSort: will run quick sort algorithm on an ArrayList
            - Partition and swap: use to help run quickSort Algorithm
        - bucketSort: will run bucketSort Algorithm on an ArrayList with a help of insertion sort
        - getDomainName: get the domain name from the URL for the use of bucketSort

5. **Class "BSTree": will store all the URLs for the specific keyword**
    - Variables
        - The root of the tree
        - searchKeyword of this BSTree
        - totalSearch of this Keyword
        - ListByIndex
        - ListByScore
        - The currentIndex of the BSTree - will never be decremented when delete a link
    - Functions
        - Constructor, Get/Set Method of all private variable
        - getArrayListByIndex: This will give the result of a sorted list by ascending Index
        - ArrayListByScore: This will set the PageRank for each URL based on their highest totalScore, and give the result of a sorted list by ascending PageRank
        - insertByScore, remove, searchByPageRank
        - CompareTo: compare each Node based on the totalSearch
6. **Class "Main": will generate the UI of the application**
    - Variables
        - All the scenes of the UI of each function of the application
        - A searchList: will store keyword and number of search of the keyword. —> this is use for display most popular search keyword
        - TableView to display data of top 30 URLs and top 10 URLs, and sorted data
        - currentSearch: to know which keyword the user is using
    - Functions
        - Many buttons will have different functions such as:
        - Each Button will have different purpose. For example,
            - When I click "**Search**" in the HomePage. It will looks if the keyword is in the HashMap or not. If no, it will create a new "Keyword" and call the WebCrawler to call Google API, and build the BSTree with 30 URLs. Then, I will use JavaFX's library to read the data from the List of the BSTree, and display into the TableView to display it on the UI. **Runtime: O(n)**
            - Click "Display Top 10": This button will loop into SearchList HashMap and create a new **Node** class that store they keyword and totalSearch of the keyword. These information can be retrieved from the BSTree. And put all the **Node** into the TableView called **Top10SearchTable. Runtime: O(n)**
            - Open **Insert**URLButton will create a new **Link** class with user's desired input of url's name, and 4 of their own factors' choice. Then, it will insert the BSTree and display the whole data of the BSTree to the TableView, **Runtime: O(h)**

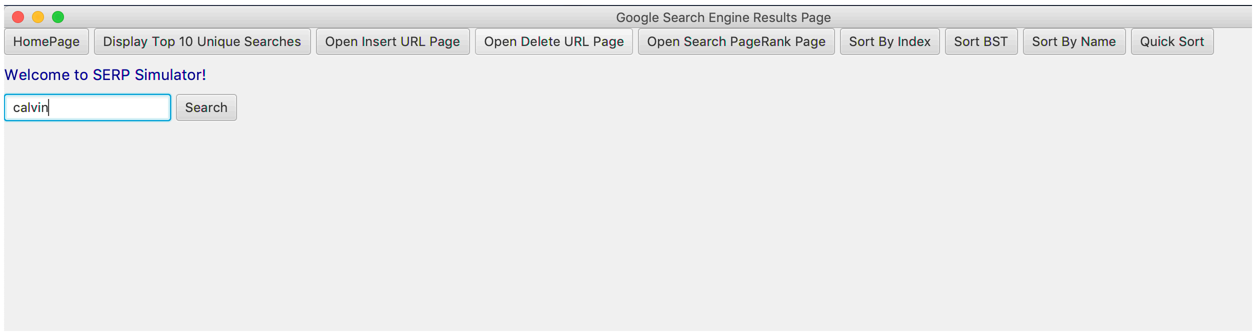
- For **Delete**URLButton and **SearchPageRank**Button, the user will be allowed to enter the pageRank they want to search or delete. When they hit enter, they will be showed the result they want to see. **Runtime: O(h)**
- SortByIndex will sort the data in the table in ascending index order, the order that they were inserted into the BSTree.
- **SortBST** will sort the data in the table in ascending PageRank order based on the highest score using BST Sort. **(Runtime: O(n^2)**
- **SortByName** will sort the data's domain name in the table in ascending alphabet order using BucketSort. **Runtime: O(n^2)**
- **QuickSort** will sort the data in the table in ascending PageRank order based on the highest score using QuickSort. **Runtime: O(n^2)**

Testing



Welcome to SERP Simulator!

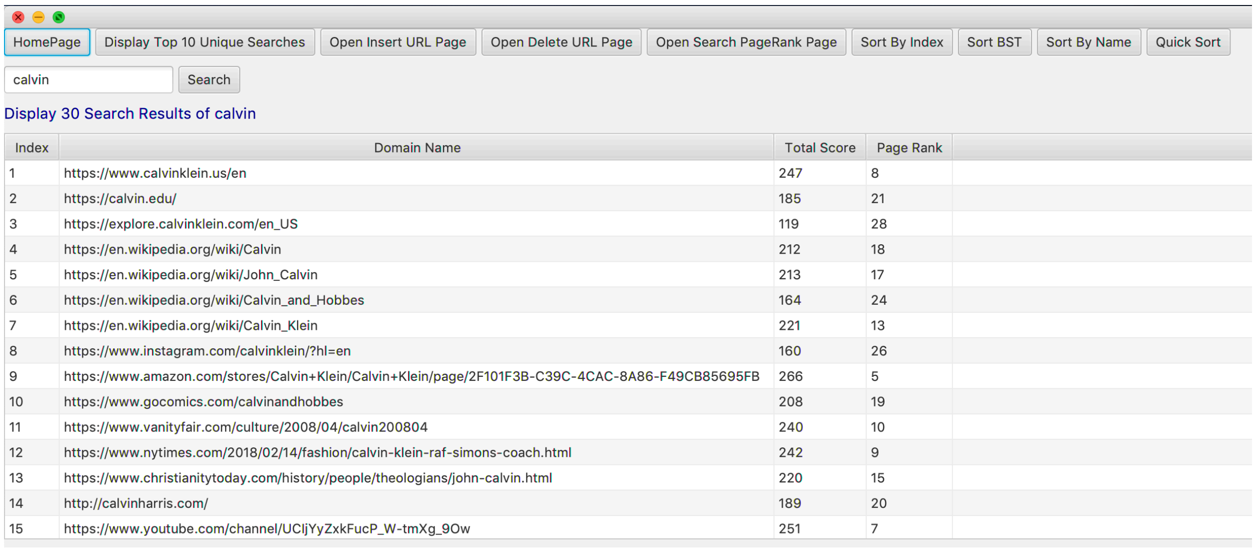
Figure 1: HomePage



Welcome to SERP Simulator!

calvin

Figure 2: Enter Search Keyword

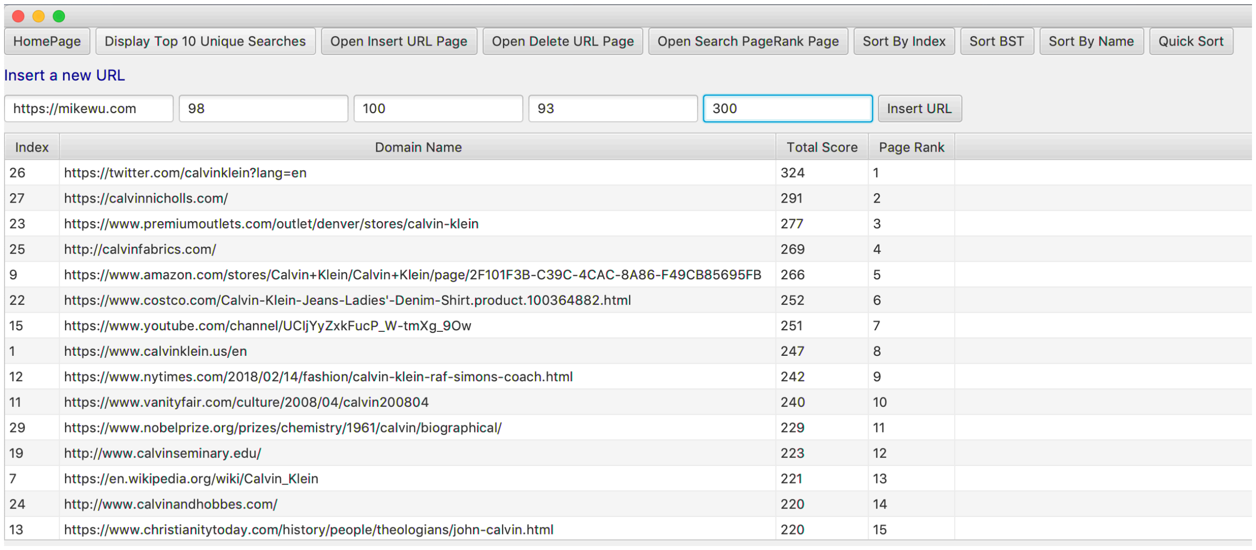


calvin

Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
1	https://www.calvinklein.us/en	247	8
2	https://calvin.edu/	185	21
3	https://explore.calvinklein.com/en_US	119	28
4	https://en.wikipedia.org/wiki/Calvin	212	18
5	https://en.wikipedia.org/wiki/John_Calvin	213	17
6	https://en.wikipedia.org/wiki/Calvin_and_Hobbes	164	24
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
8	https://www.instagram.com/calvinklein/?hl=en	160	26
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
10	https://www.gocomics.com/calvinandhobbes	208	19
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15
14	http://calvinharris.com/	189	20
15	https://www.youtube.com/channel/UCIjYyZxkFucP_W-tmXg_9Ow	251	7

Figure 3: Display 30 Results. Please scroll down when you open the file

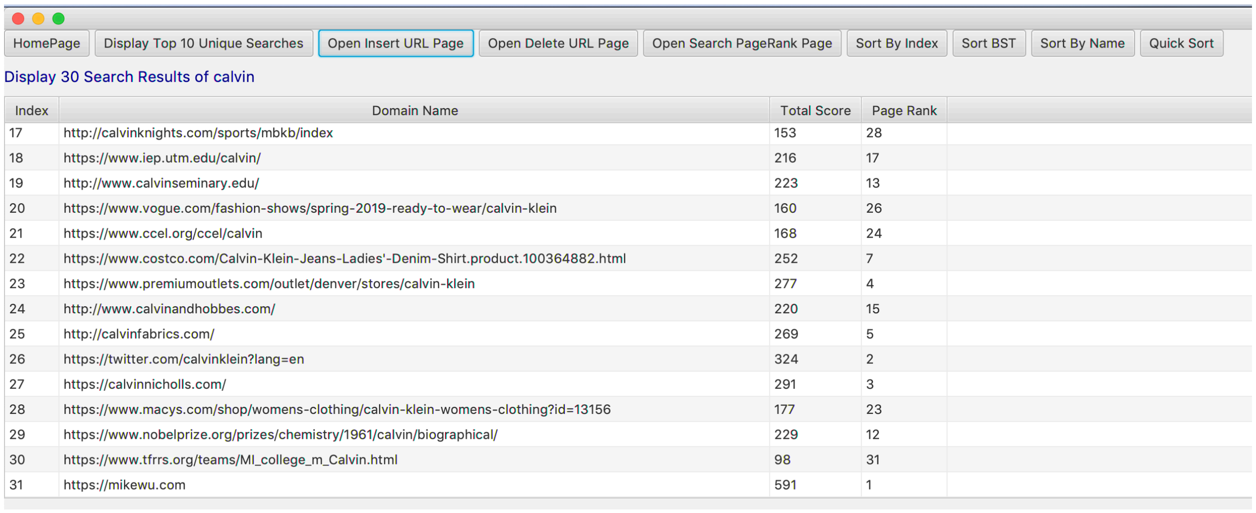


Insert a new URL

https://mikewu.com | 98 | 100 | 93 | 300

Index	Domain Name	Total Score	Page Rank
26	https://twitter.com/calvinklein?lang=en	324	1
27	https://calvinnicholls.com/	291	2
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	3
25	http://calvinfabrics.com/	269	4
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	6
15	https://www.youtube.com/channel/UCIjYyZxkFucP_W-tmXg_9Ow	251	7
1	https://www.calvinklein.us/en	247	8
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	11
19	http://www.calvinseminary.edu/	223	12
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
24	http://www.calvinandhobbes.com/	220	14
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15

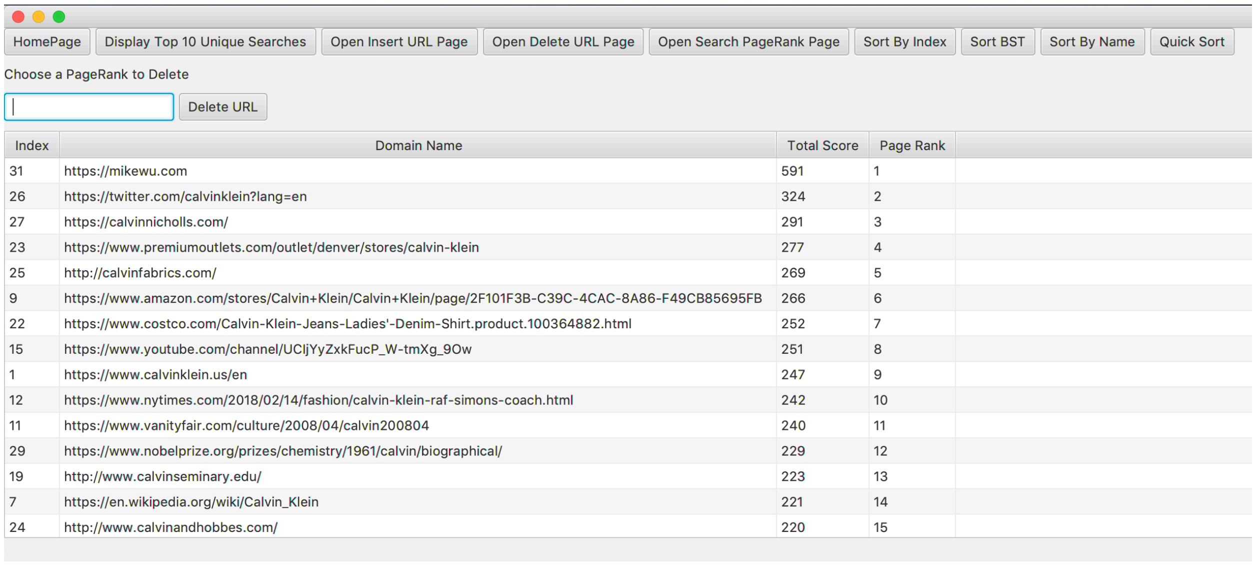
Figure 4: Insert a new URL with 4 factors of user's choice



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
17	http://calvinknights.com/sports/mbkb/index	153	28
18	https://www.iep.utm.edu/calvin/	216	17
19	http://www.calvinseminary.edu/	223	13
20	https://www.vogue.com/fashion-shows/spring-2019-ready-to-wear/calvin-klein	160	26
21	https://www.ccel.org/ccel/calvin	168	24
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	7
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
24	http://www.calvinandhobbes.com/	220	15
25	http://calvinfabrics.com/	269	5
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
28	https://www.macys.com/shop/womens-clothing/calvin-klein-womens-clothing?id=13156	177	23
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	12
30	https://www.tfrrs.org/teams/MI_college_m_Calvin.html	98	31
31	https://mikewu.com	591	1

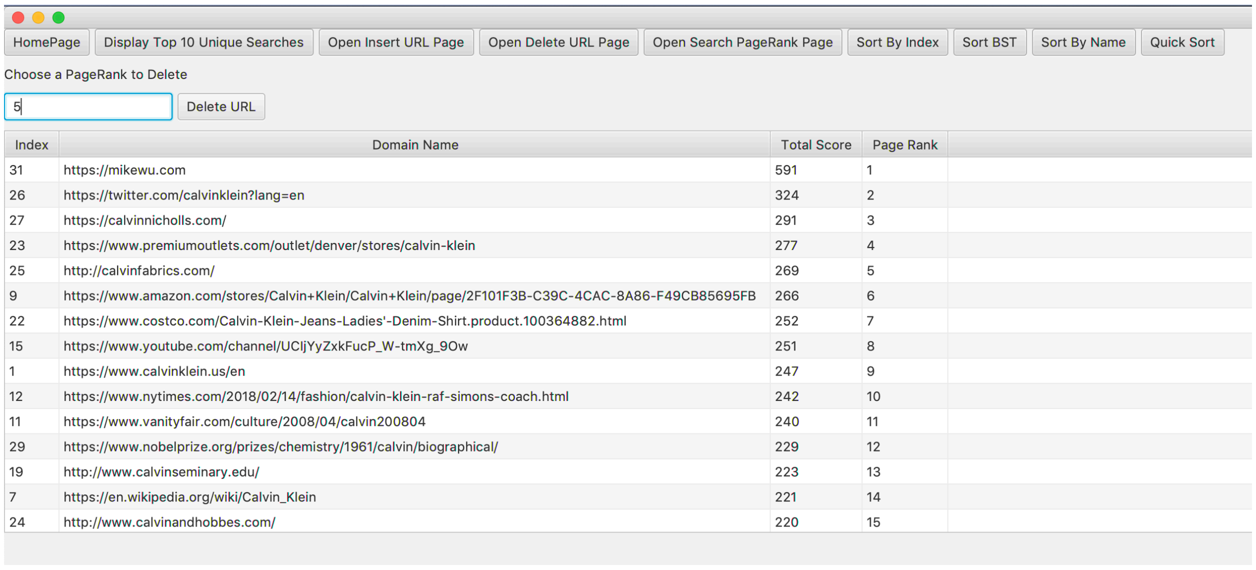
Figure 5: Display the result at index 31 after inserting new URL



Choose a PageRank to Delete

Index	Domain Name	Total Score	Page Rank
31	https://mikewu.com	591	1
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
25	http://calvinfabrics.com/	269	5
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	6
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	7
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	8
1	https://www.calvinklein.us/en	247	9
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	10
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	11
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	12
19	http://www.calvinseminary.edu/	223	13
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	14
24	http://www.calvinandhobbes.com/	220	15

Figure 6: Choose a PageRank to delete

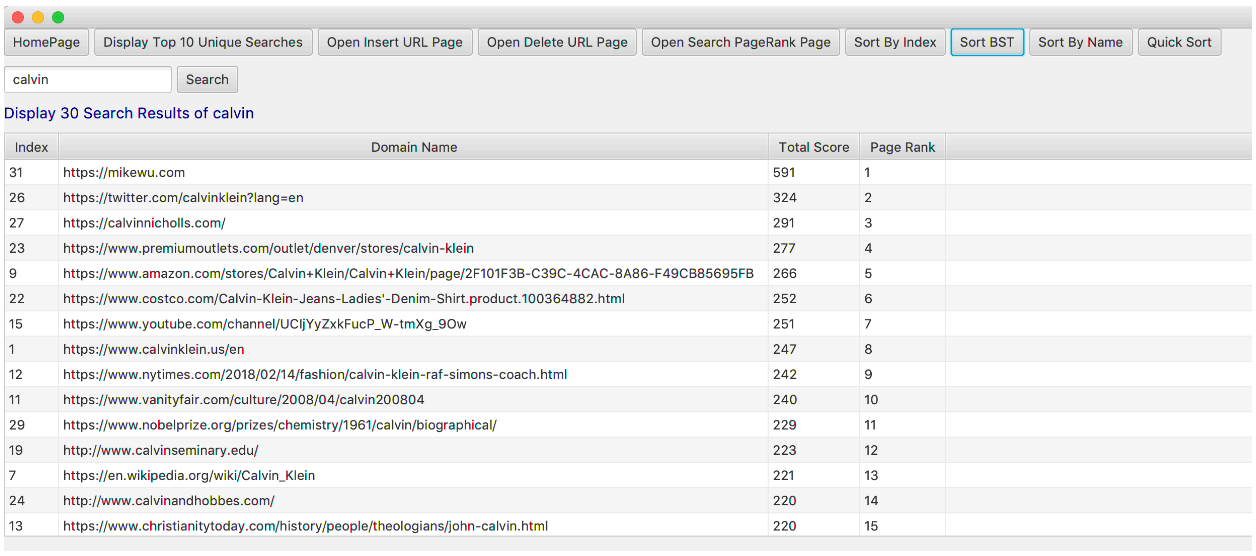


Choose a PageRank to Delete

5

Index	Domain Name	Total Score	Page Rank
31	https://mikewu.com	591	1
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
25	http://calvinfabrics.com/	269	5
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	6
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	7
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	8
1	https://www.calvinklein.us/en	247	9
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	10
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	11
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	12
19	http://www.calvinseminary.edu/	223	13
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	14
24	http://www.calvinandhobbes.com/	220	15

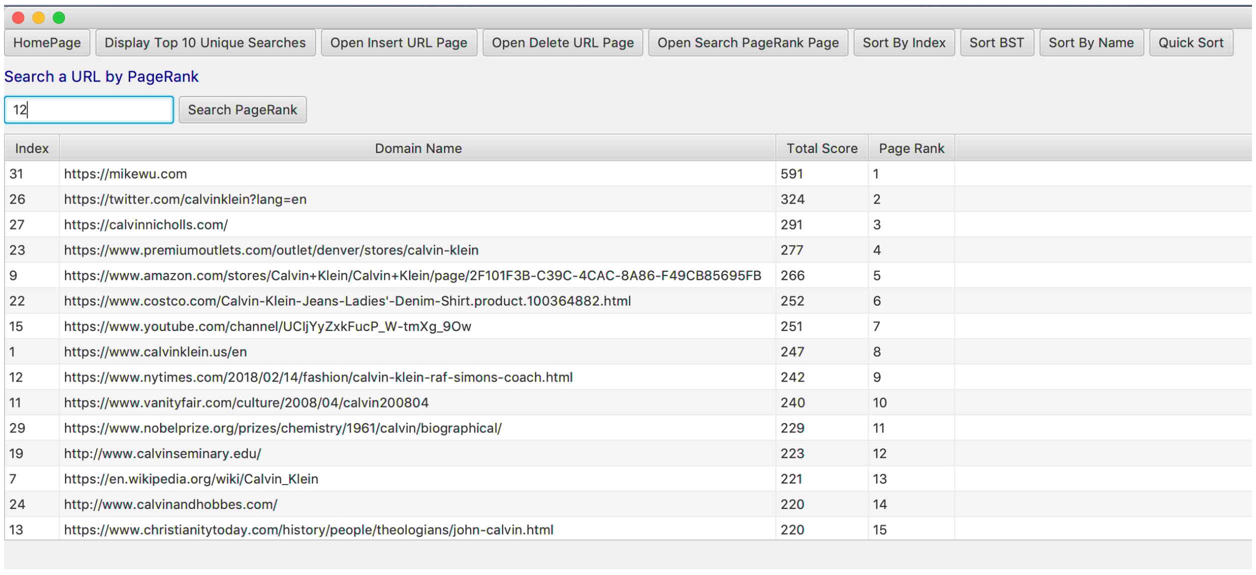
Figure 7: Choose PageRank 5 to delete



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
31	https://mikewu.com	591	1
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	6
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	7
1	https://www.calvinklein.us/en	247	8
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	11
19	http://www.calvinseminary.edu/	223	12
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
24	http://www.calvinandhobbes.com/	220	14
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15

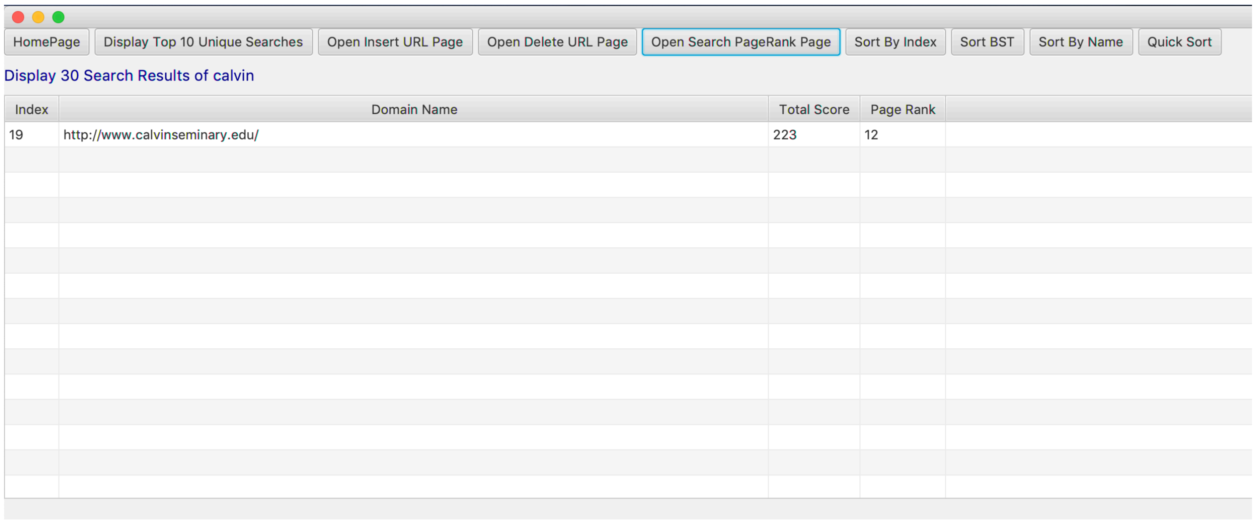
Figure 8: Show Result after delete PageRank 5



Search a URL by PageRank

Index	Domain Name	Total Score	Page Rank
31	https://mikewu.com	591	1
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	6
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	7
1	https://www.calvinklein.us/en	247	8
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	11
19	http://www.calvinseminary.edu/	223	12
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
24	http://www.calvinandhobbes.com/	220	14
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15

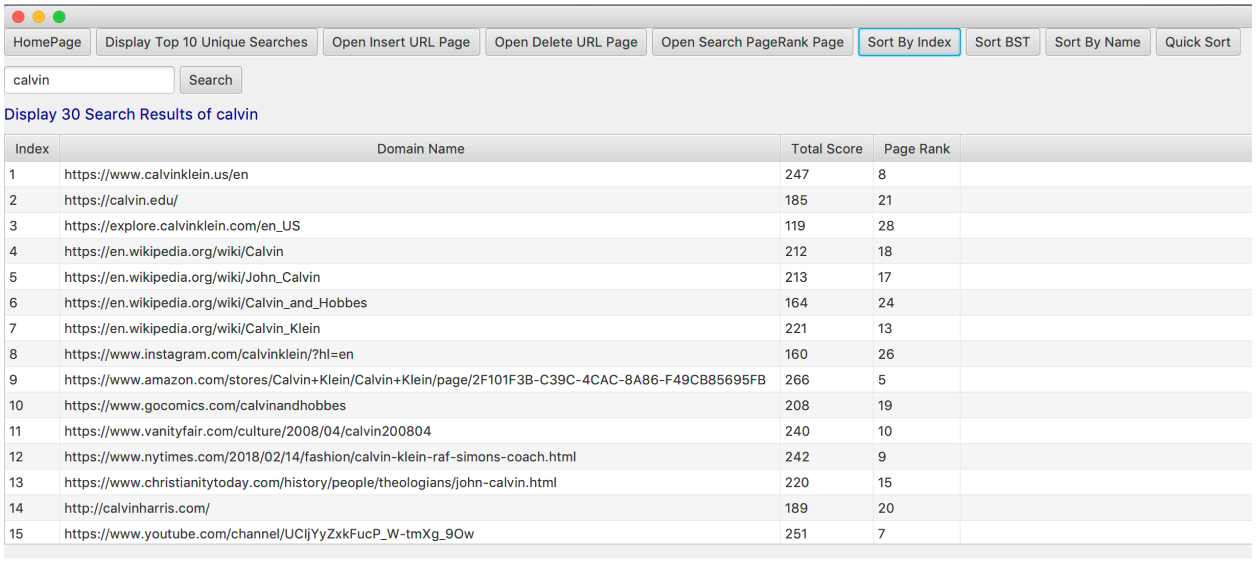
Figure 9: Search PageRank 12



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
19	http://www.calvinseminary.edu/	223	12

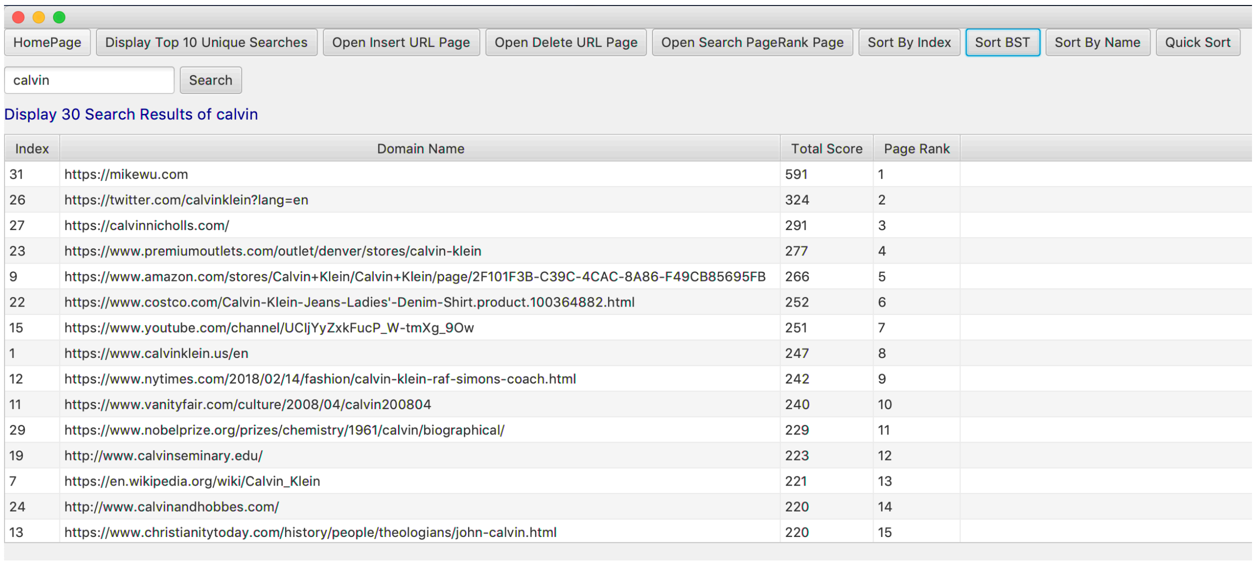
Figure 10: Display Full Result of PageRank 12



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
1	https://www.calvinklein.us/en	247	8
2	https://calvin.edu/	185	21
3	https://explore.calvinklein.com/en_US	119	28
4	https://en.wikipedia.org/wiki/Calvin	212	18
5	https://en.wikipedia.org/wiki/John_Calvin	213	17
6	https://en.wikipedia.org/wiki/Calvin_and_Hobbes	164	24
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
8	https://www.instagram.com/calvinklein/?hl=en	160	26
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
10	https://www.gocomics.com/calvinandhobbes	208	19
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15
14	http://calvinharris.com/	189	20
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	7

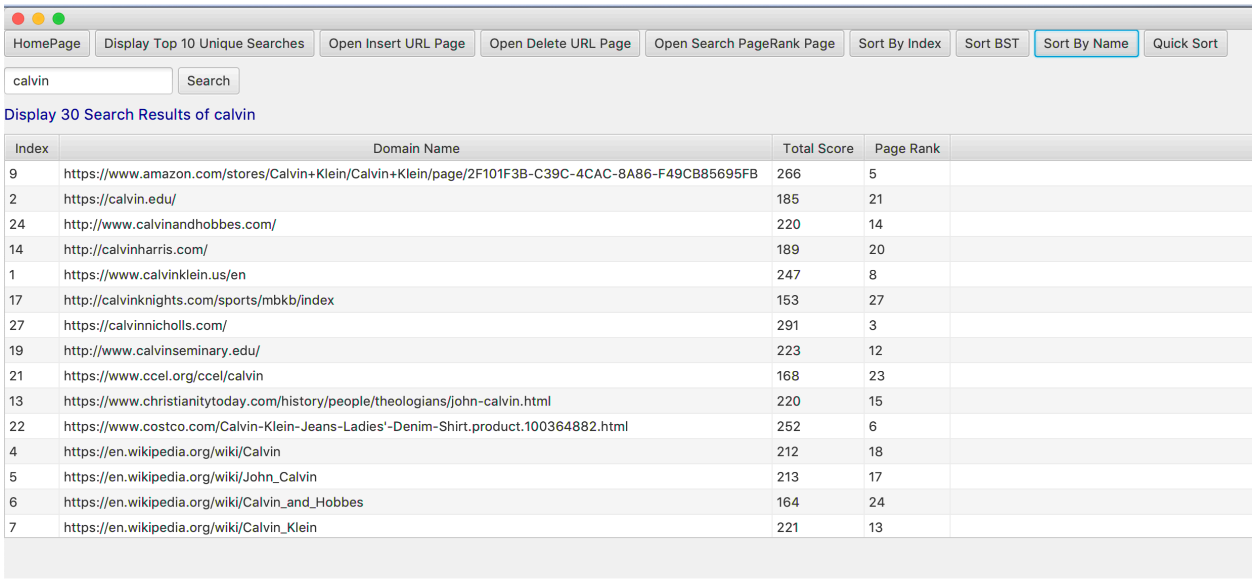
Figure 11: Sort the Data by Ascending Index



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
31	https://mikewu.com	591	1
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	6
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	7
1	https://www.calvinklein.us/en	247	8
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	11
19	http://www.calvinseminary.edu/	223	12
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
24	http://www.calvinandhobbes.com/	220	14
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15

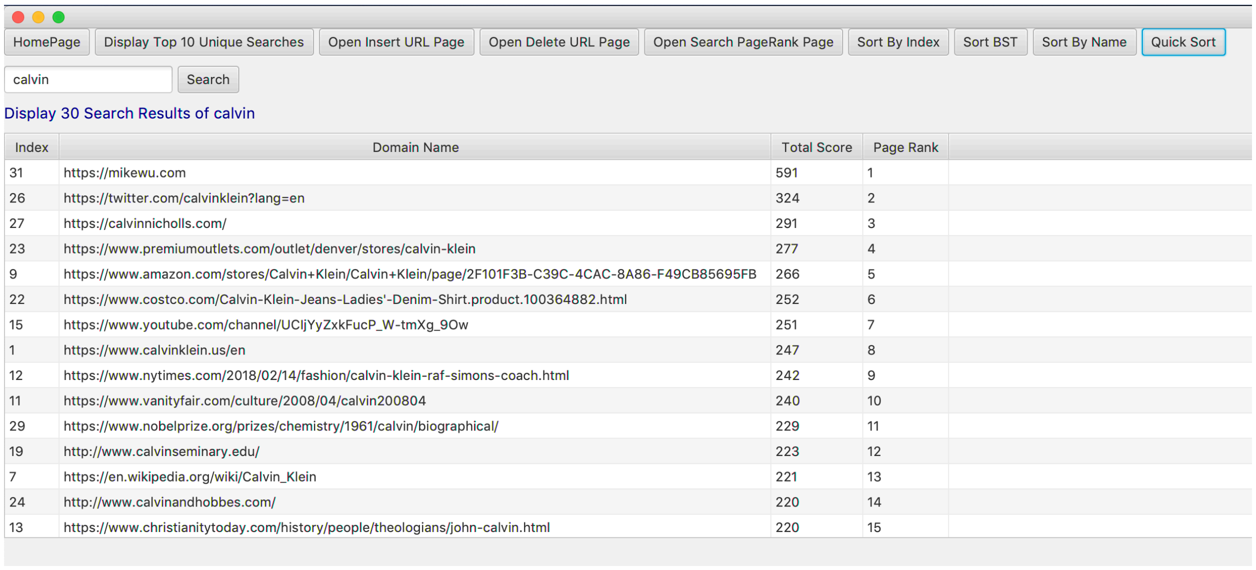
Figure 12: Sort the Data By PageRank using BST Sort



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
2	https://calvin.edu/	185	21
24	http://www.calvinandhobbes.com/	220	14
14	http://calvinharris.com/	189	20
1	https://www.calvinklein.us/en	247	8
17	http://calvinknights.com/sports/mbkb/index	153	27
27	https://calvinnicholls.com/	291	3
19	http://www.calvinseminary.edu/	223	12
21	https://www.ccel.org/ccel/calvin	168	23
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	6
4	https://en.wikipedia.org/wiki/Calvin	212	18
5	https://en.wikipedia.org/wiki/John_Calvin	213	17
6	https://en.wikipedia.org/wiki/Calvin_and_Hobbes	164	24
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13

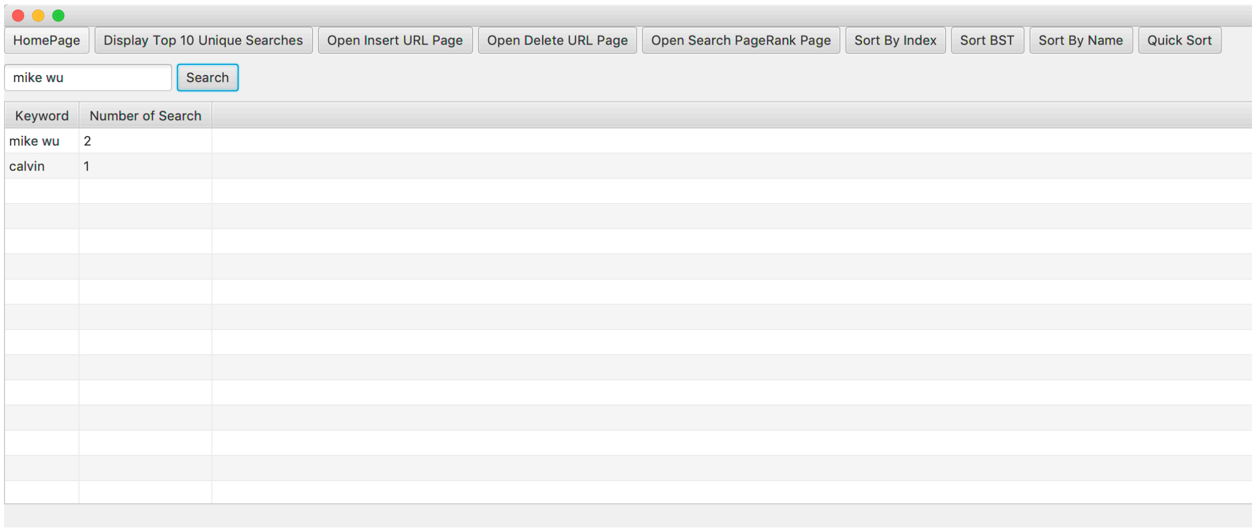
Figure 13: Sort the current Search By Name using BucketSort



Display 30 Search Results of calvin

Index	Domain Name	Total Score	Page Rank
31	https://mikewu.com	591	1
26	https://twitter.com/calvinklein?lang=en	324	2
27	https://calvinnicholls.com/	291	3
23	https://www.premiumoutlets.com/outlet/denver/stores/calvin-klein	277	4
9	https://www.amazon.com/stores/Calvin+Klein/Calvin+Klein/page/2F101F3B-C39C-4CAC-8A86-F49CB85695FB	266	5
22	https://www.costco.com/Calvin-Klein-Jeans-Ladies'-Denim-Shirt.product.100364882.html	252	6
15	https://www.youtube.com/channel/UCljYyZxkFucP_W-tmXg_9Ow	251	7
1	https://www.calvinklein.us/en	247	8
12	https://www.nytimes.com/2018/02/14/fashion/calvin-klein-raf-simons-coach.html	242	9
11	https://www.vanityfair.com/culture/2008/04/calvin200804	240	10
29	https://www.nobelprize.org/prizes/chemistry/1961/calvin/biographical/	229	11
19	http://www.calvinseminary.edu/	223	12
7	https://en.wikipedia.org/wiki/Calvin_Klein	221	13
24	http://www.calvinandhobbes.com/	220	14
13	https://www.christianitytoday.com/history/people/theologians/john-calvin.html	220	15

Figure 14: Sort the data by ascending PageRank using QuickSort



Keyword	Number of Search
mike wu	2
calvin	1

Figure 15: Show the popular search by the user

Challenges & Lesson Learned

Challenges

In this project, the most challenging problem that I encountered and spent the most time on is making the User Interface as well as fetching the data and posting the data between front-end and backend. It's so hard that even when I knew that the backend got the right answer, but the UI doesn't show the right data. Hence, I had to spend a lot of time to print out the data in the console and debugger to see which line is not working because it is not algorithm, it's just problem with syntax and I have never worked with JavaFX, the tool to make UI in Java. Especially, when I tried to call the Google API too many times by using JSOUP Library, Google actually blocked my IP address to prevent from hacking. Hence, I had to make sure everything is right before I run the program otherwise I had to wait for one day to continue work on the project.

Lesson Learned

After this project and previous project, my experience in building the User Interface is getting better and better. I know what components are supposed to be built first. Regardless of the limitation of using Google API, I could practice more in working with the code before running it. It helps me to develop my analytic skills as I need to know if my code is making sense or not. Especially, I learned how to integrate BSTree Data Structure, learned different class design and implementation as well as how to use QuickSort and BucketSort in real life application.