**Google Search Engine Simulation**

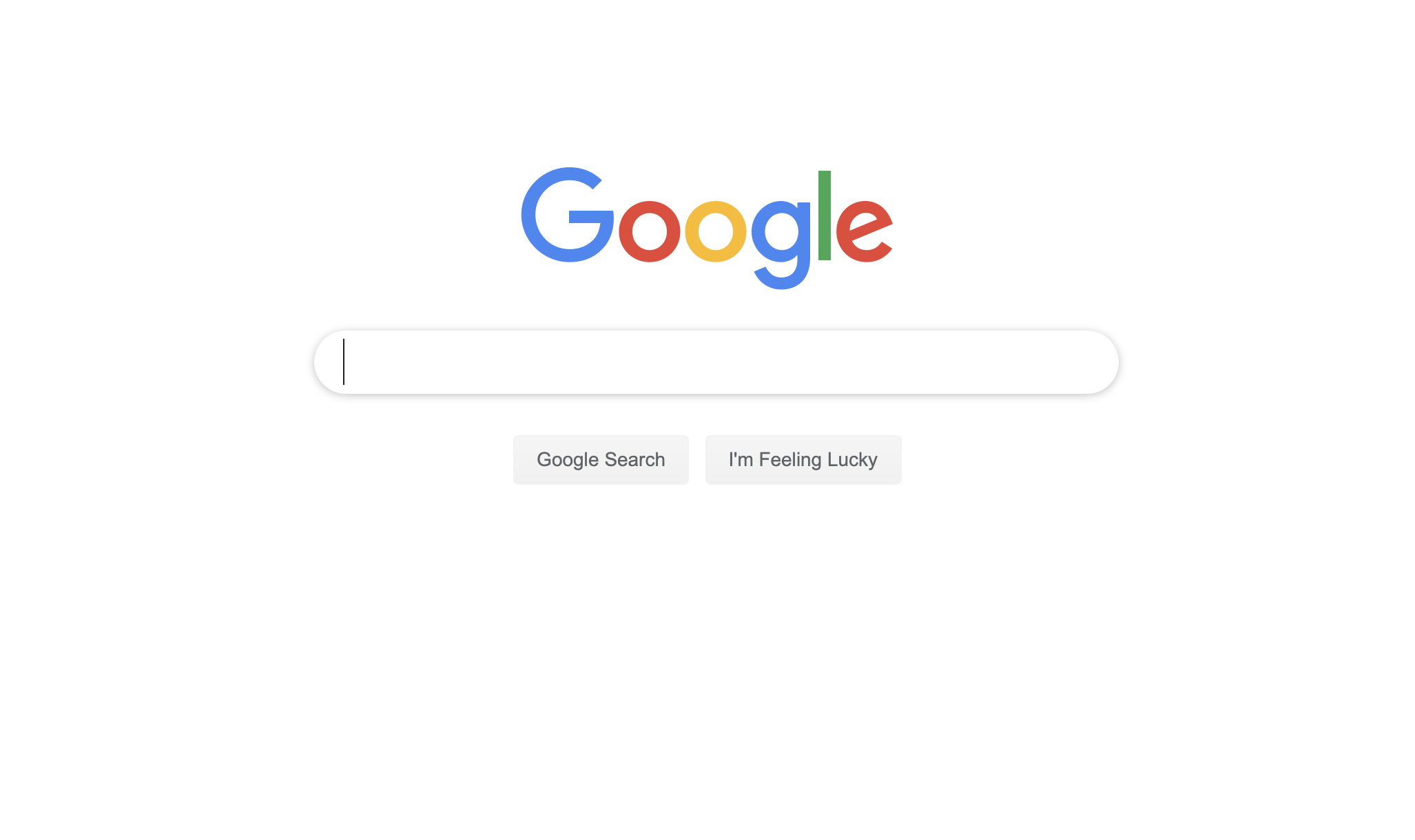
Designer: Zehui Zhan

Date: March 24, 2019

Institution: San Jose State University

Adviser: Professor Wu

Course: Data Structures and Algorithms, CS146, Spring 2019



***Introduction:***

Search engine is a powerful system and the Google search engine is the most popular one around the word. It could search all relevancy thing online by the key word. It also uses different sorting algorithm to sort the relevancy thing, rank by few factors and present the most useful link for you. In this project, I was trying to make a Google search engine simulation. Two major function I implemented was Heapsort algorithm and Max-Heap Priority Queue. The user could type into a key word for this Google search engine simulation and it would generate a list of links which was relevancy with the key word. For searching the key word on the internet, I implemented the Web Crawler and it would do help to search it in the google and return a list of links. What’s more, the trademarked algorithm called PageRank was designed for ranking the relevancy score for each link. The result for list of links would be presented to user which ranks by the relevancy score. In order to make it, I used the Heapsort algorithm to sort the relevancy score. If user would like to insert a new link, this Google search engine simulation also can make it. The score for it would be re-rank and re-present in the list. However, if the page’s owner would like to pay more money to the user to increase the page rank for advertisement purpose, this simulation also can make it. The score for every link could be changed and the page rank would be re-arranging automatically. That is the major function for this simulation.

***Table of Contents***

Introduction……………………………………………………………...………………………...1

Table of Contents………………………………………………………………………………….2

Design and Implementation………………………………………………………………….…3-6

A list of classes/subroutines/function calls…………………………………………….….……7-8

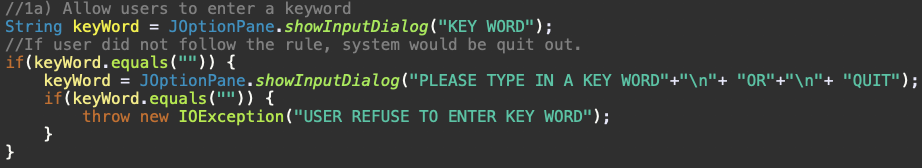
Self-testing screen shots…………………………………………………………………….…9-12

The procedure (step by step) ……………………………………………………………………13

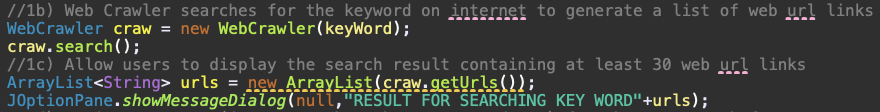
Problems encountered………………………………………………...……………………..14-15

Lessons Learned………………………………………………………………………….………16

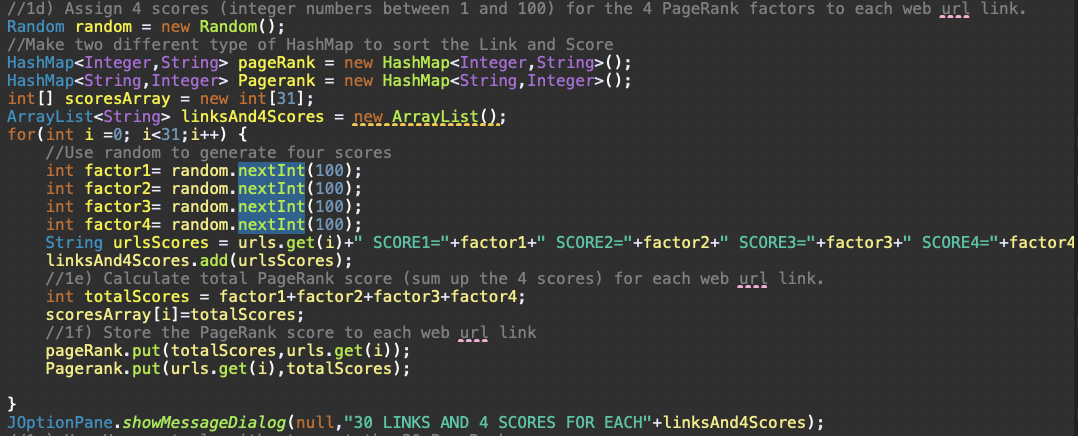
***Design and Implementation***

The whole project designment based on Heapsort algorithm and Max-heap Priority Queue. For completing this two major method, I did seven sub method for them first. What’s more, I implemented the WebCrawler class to collect the relative links from google.

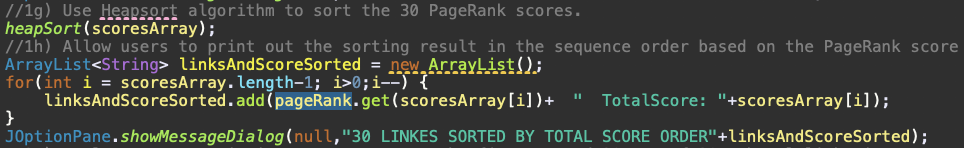
At the beginning of the main method, I used the JOptionPane to get the key word from the user. The way I had learned to do it was Scanner right now, but I found out that JOptionPane is more professional and efficient than it. If user forget to type in a key word, the system would give him a second chance to do it. But if user still not try to type in a key word, it would be counted as the user refuse to entry key word and automatically quit out.



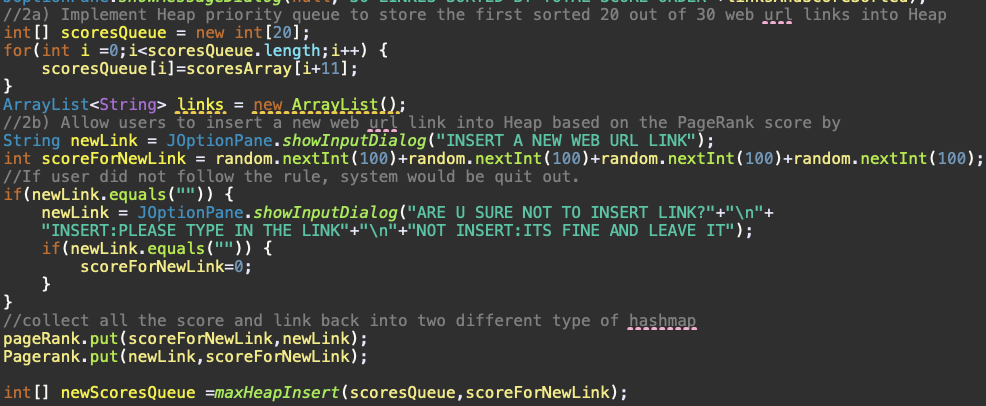
When the key word gets into the system. The WebCrawler class could accept it and call the search method to search the key word in Google and return the links. The thing I used to collect the links is ArrayList because it is easy to print out and arrange. The getUrls would be called and use JoptionPane.showMessageDialog to print out this ArrayList and present all the links which get from Google to user.



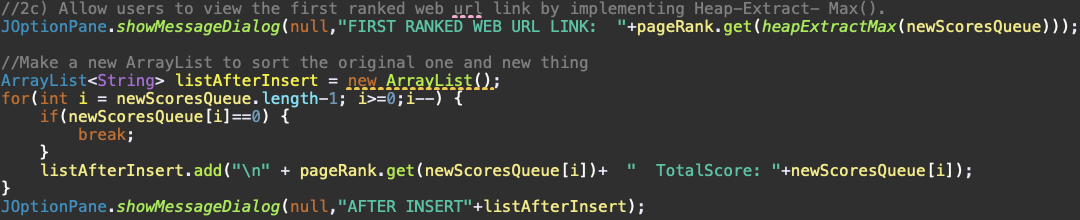
For the relevancy score, I used the Random to generate four score which represent four factors. I used the HashMap to collect the links and the score for each links. That’s is easy to get them back. I summed up four scores for each links and put them into an array, which it’s the start of the heap. Then I show the user with the 30 links and 4 scores for each.



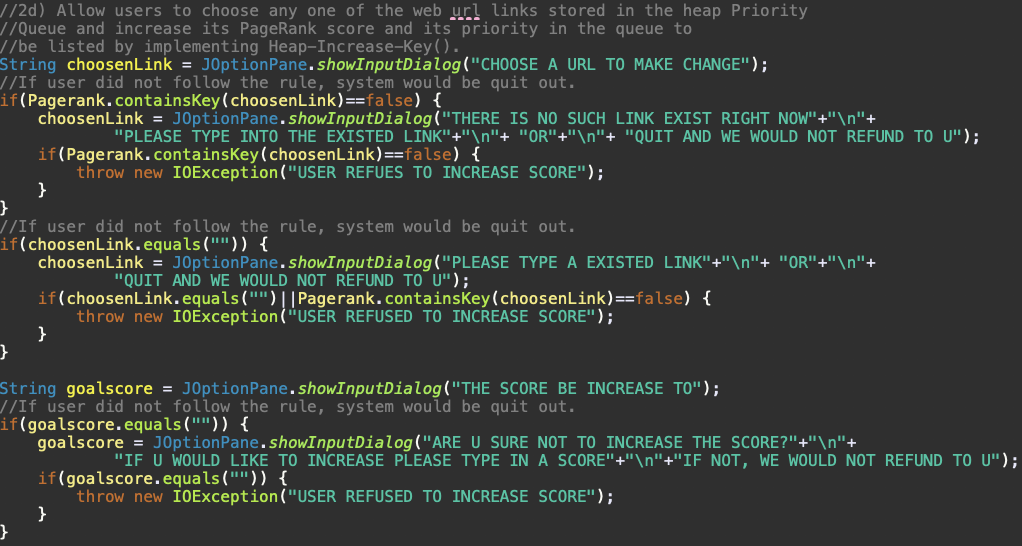
The heap array I called scoresArray and I used heapsort method to sort it. I made a new ArrayList and it collect the right rank links with the total score for each by the order of the scoresArray. The link I found back by the get method of the HashMap, find the value by key. Since the result for heapsort is from the smaller to bigger and what we need to present to user is from high relevancy scores to low, I used the for loop and start with the tail of the scoresArray. Then I show the user with the ranked 30 links and the total score for each.



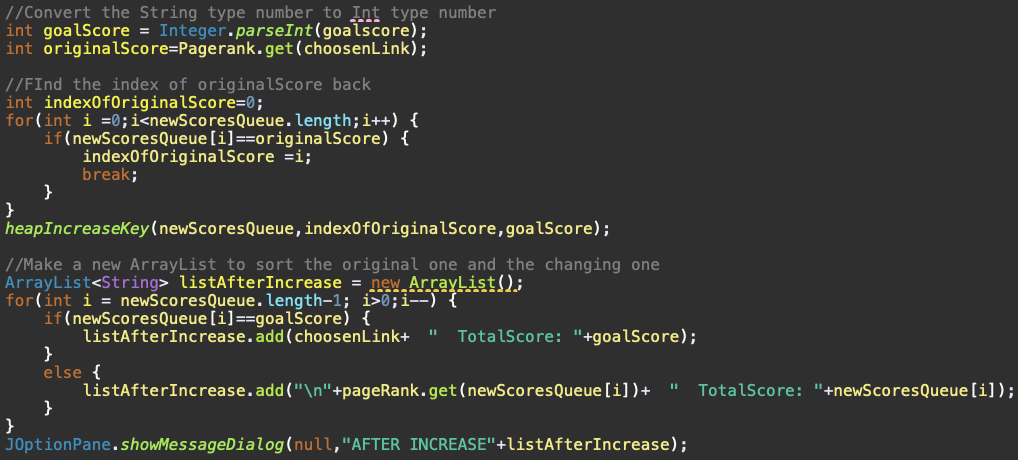
I made an new array to store the first sorted 20 out of 30 links and allowed user to insert a new links. If user did not type in the web links, the system would double check with him and ask him again. If user still do not type in a link, the system would skip for it and nothing would be disturbed. If user do type in a link, the system would generate a total relevancy score for it. The new link and new score would be stored into the HashMap pageRank and the score for new link would be store into the array by calling the maxHeapInsert Method.



The first ranked web link would be presented to user by calling the heapExtractMax method and find the required link. After showing the first web link to user, the list of the web link with the total score for each would appear. The inserted link also would be present in the links (except it was extracted).



After inserting a new link, the user could increase the relevancy scores for the exited link. System would allow user to type in an exited link but if user did not type in or type in an not exited link, program would double check with user. If the user did not type in a correct link, the system would be quit out automatically. Same as the given goal score.



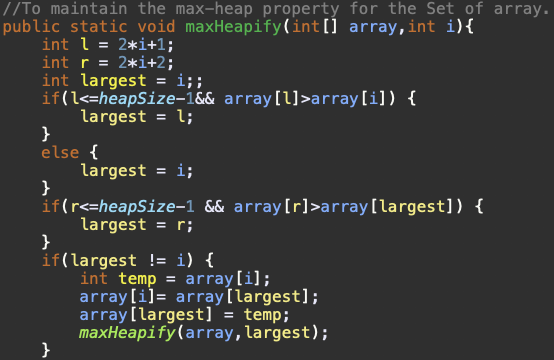
The index of the original score would be find out and call the heapIncreaseKey to change it to goal score. The list after increasing score would be presented to the user.

***A list of classes/subroutines/function calls***

* WebCrawler class

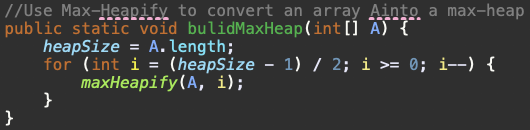
To search the key word in google and return the relevancy link.

* maxHeapify



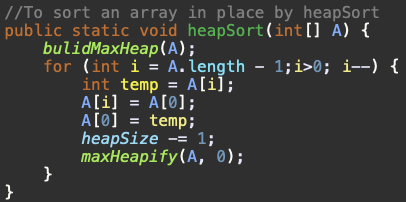
To maintain the max-heap property for the Set of arrays. (Parent node must not bigger than the child node, if is, swap them)

* bulidMaxHeap

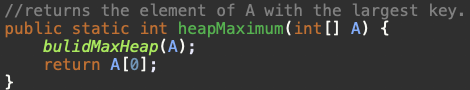


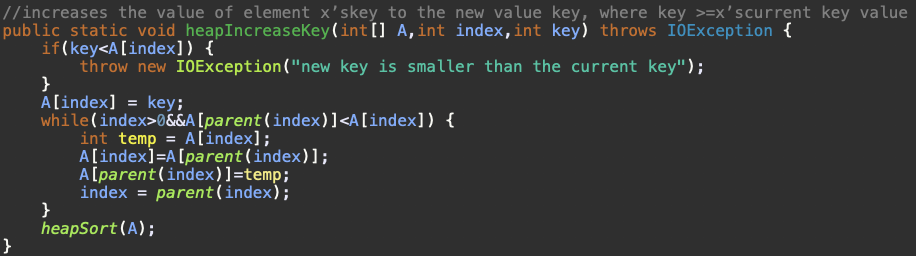
Use maxHeapify method to convert an array in to a max-heap.

* heapSort()

A way to sort an array in place by using the blidMaxHeap method and maxHeapify method. The result of the array is from smaller value to bigger value.

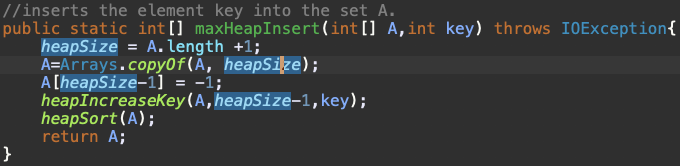
* heapMaximum

Return the largest value of the array. Since the array is max heap, the largest value is in the index of 1.

* heapIncreaseKey

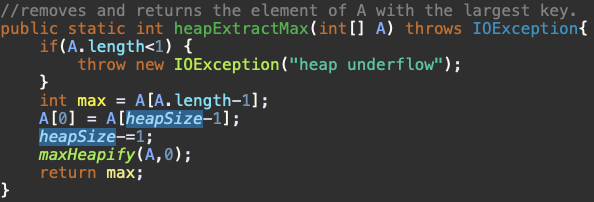
Increaess the value of element x’s key to the new value key, where key bigger or equals to the current key value.

* maxHeapInsert

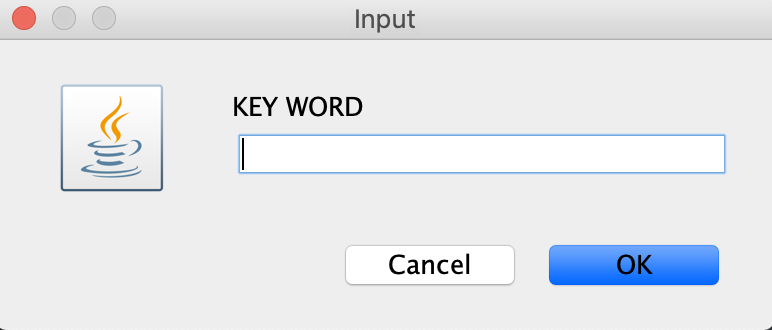
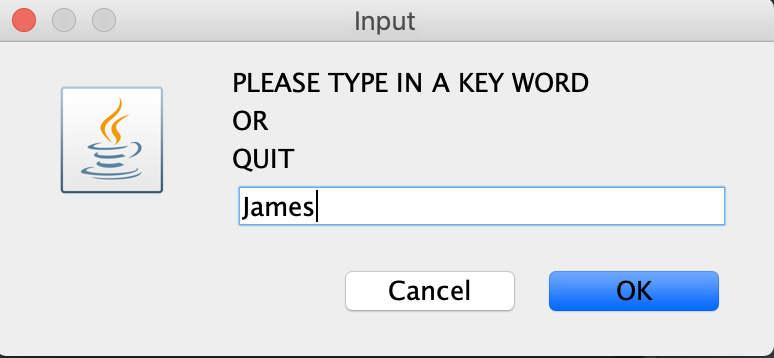


Insert an element key in to the set A and keep the order.

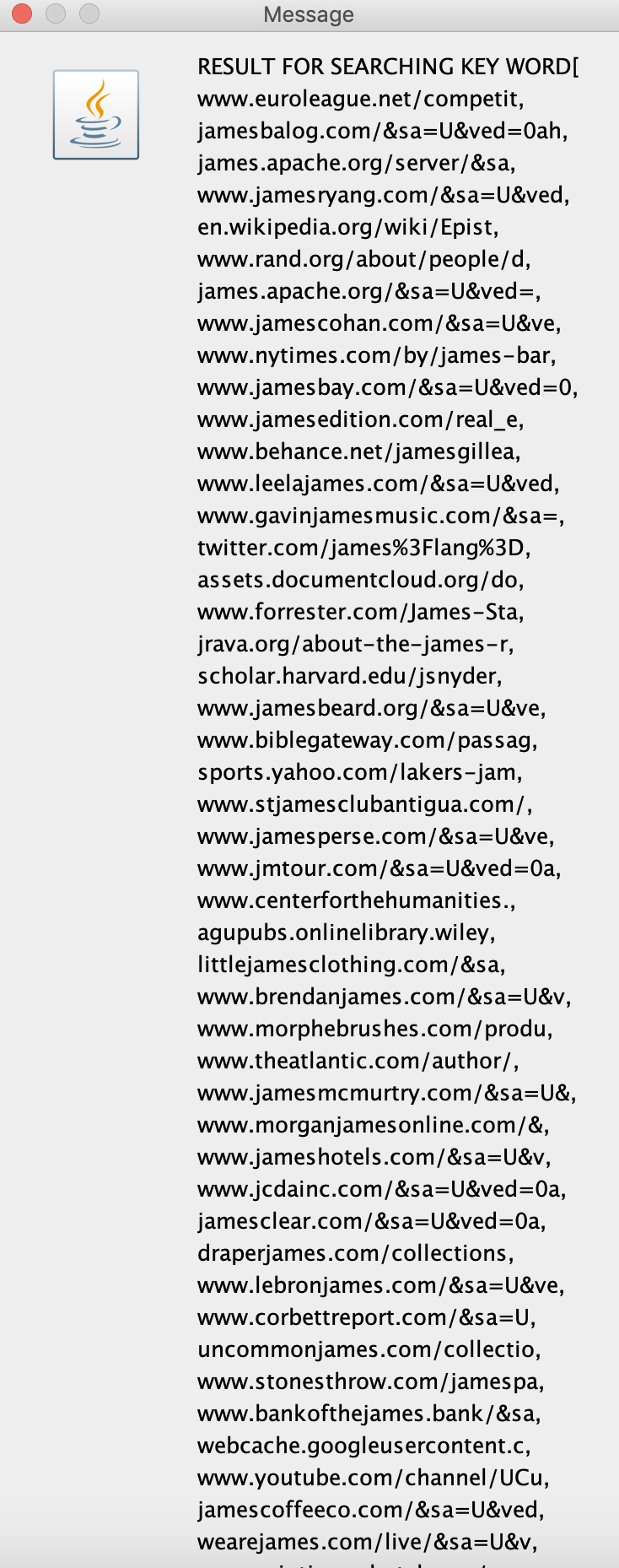
* heapExtracMax

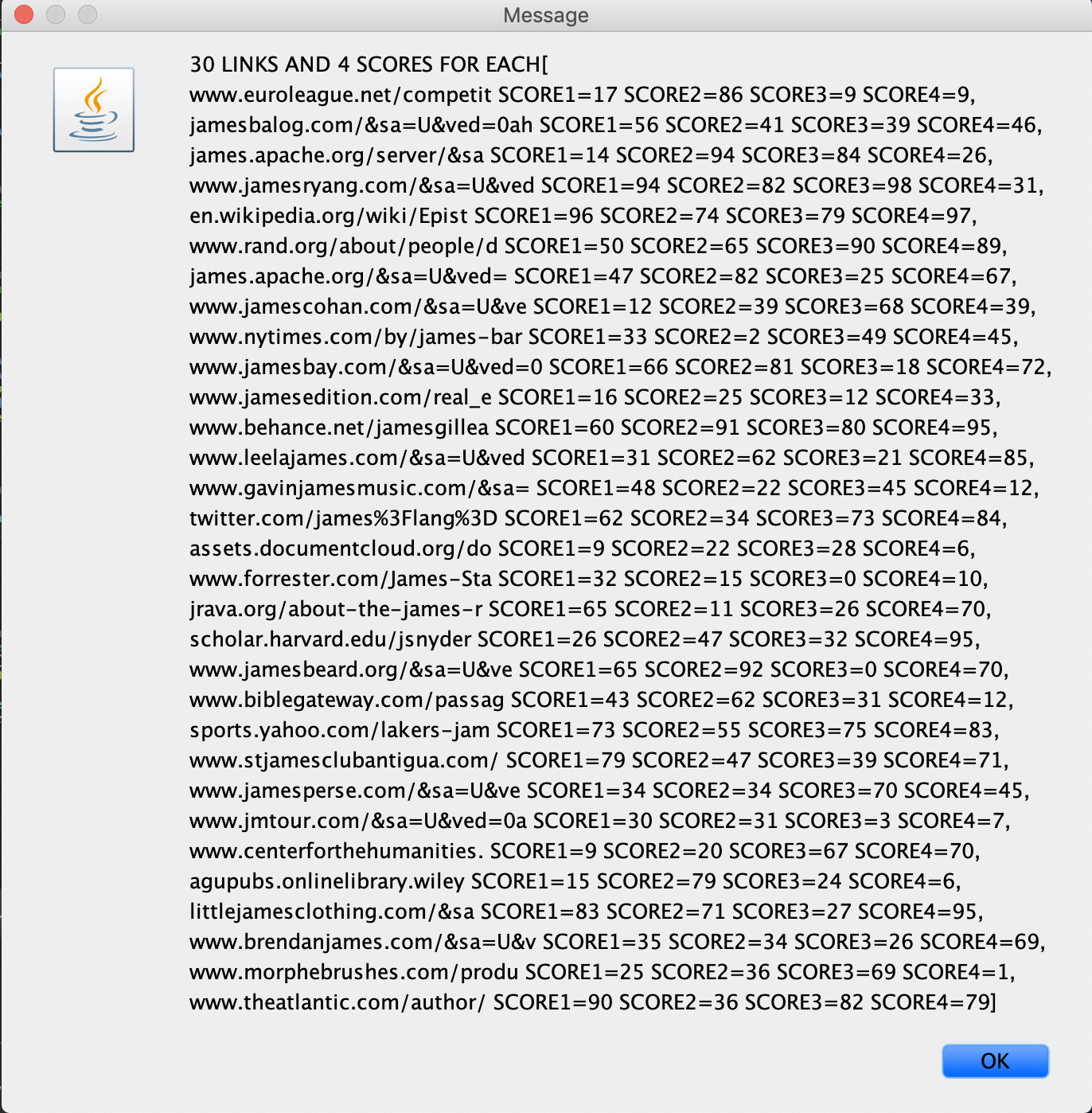
Removes and returns the element of A with the largest key. Removes the largest value first and keep the array in order.

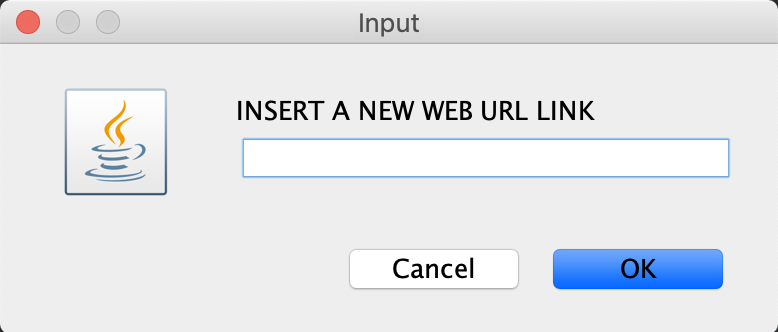
***Self-testing screen shots***

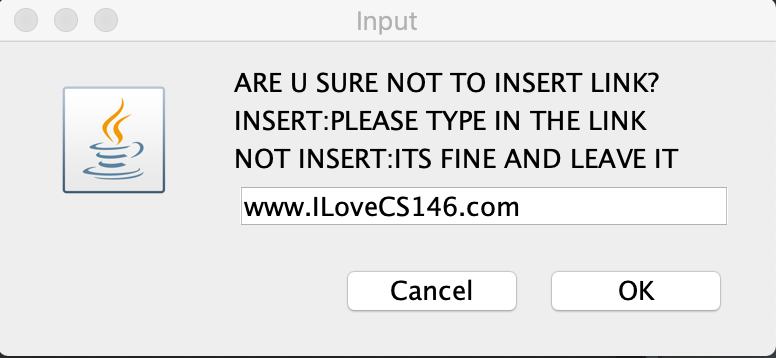
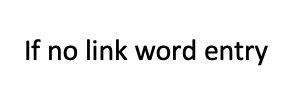


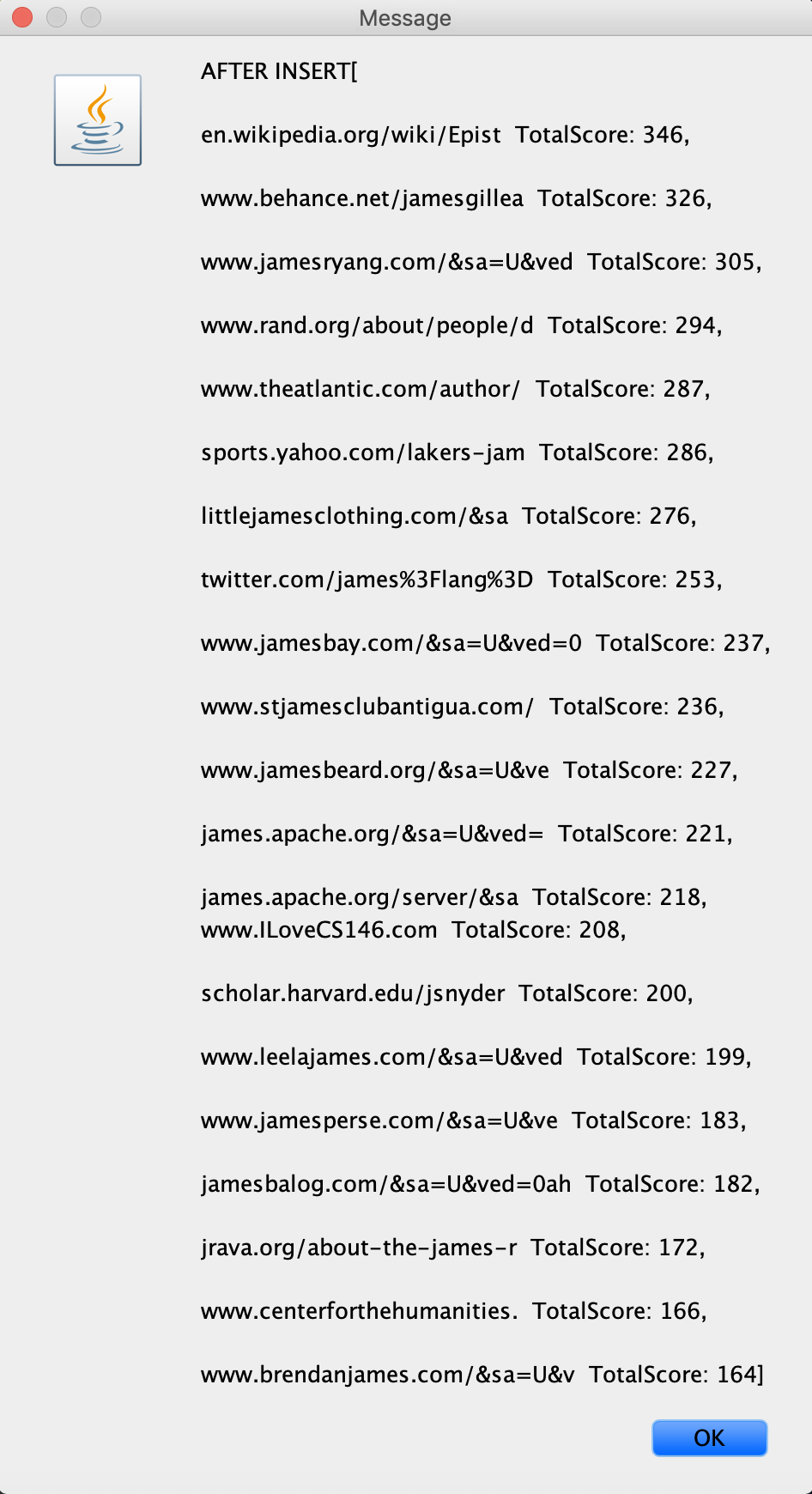
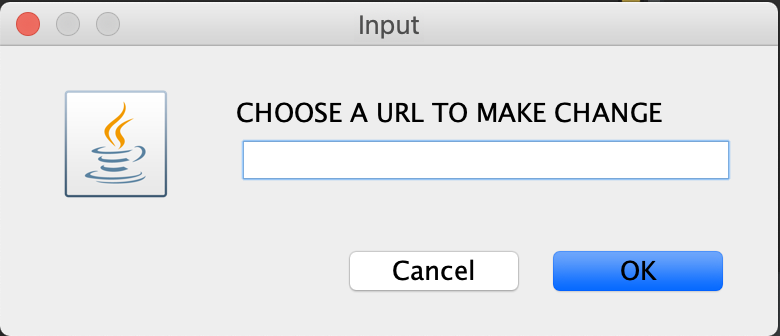
If no key word entry

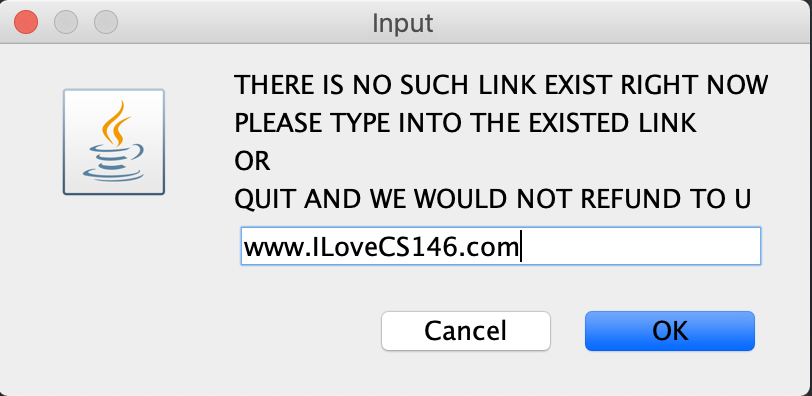
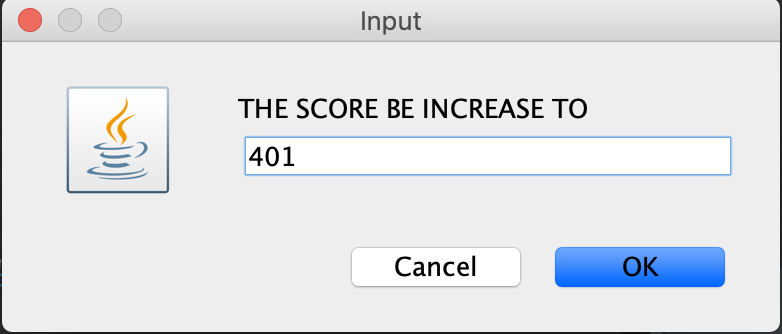
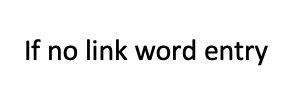


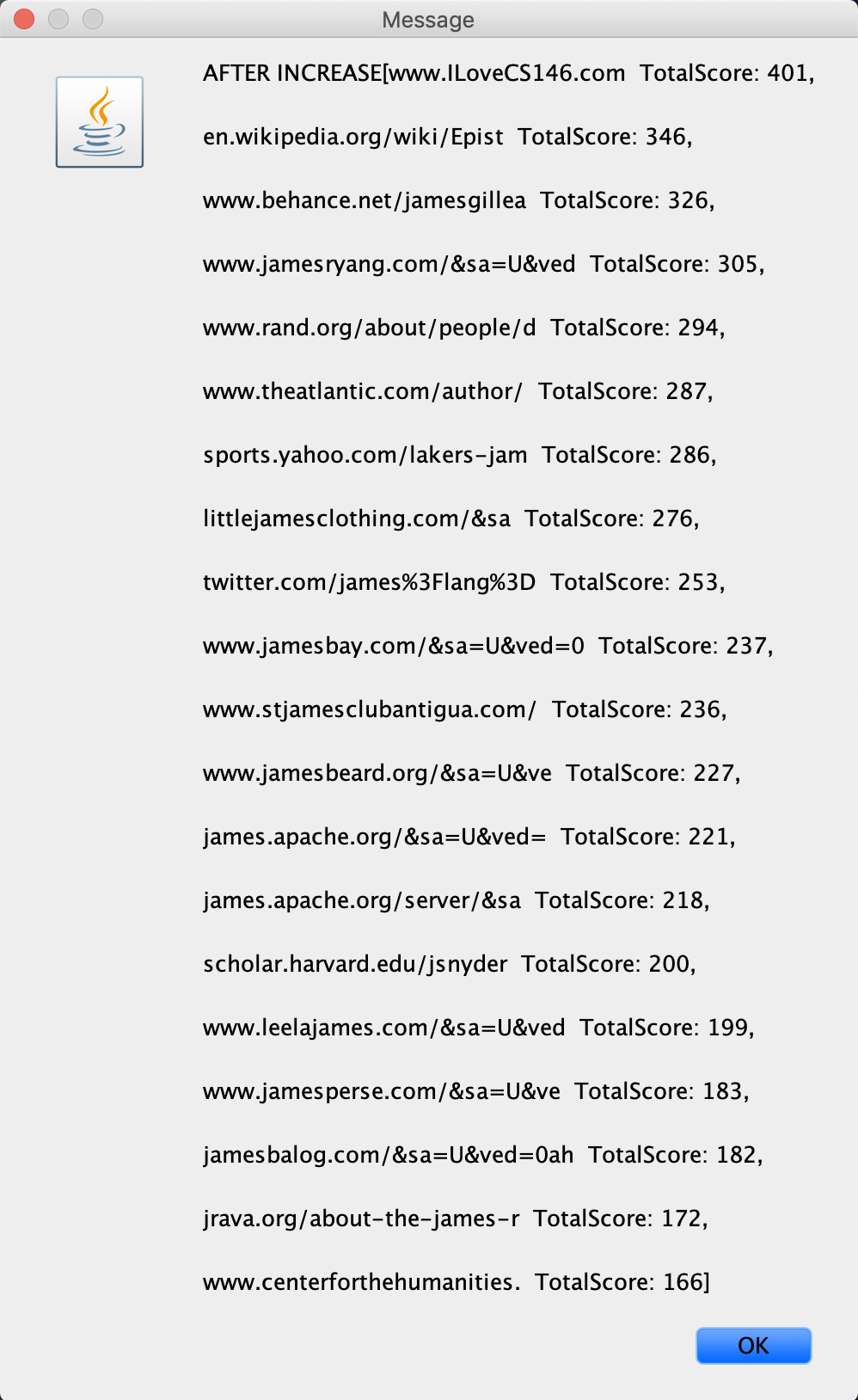












***The procedure (step by step)***

* unzip the file in a normal way
* open the application Eclipse or any Java application
* import two classes (Simulation and WebCrawler) in the file to Eclipse
* click the running button on the top (green arrow)
* enter a key word you would like to test
* (showing the result of the link and score)
* enter a new link you would like to insert if you like
* (showing the result of the link and score after insert)
* enter an exited link
* enter a score you would like to increase to
* (showing the result of the link and score after increase one of the scores)

**\* The system accept that the user did not enter the link or score, which could also be testing by trying to leave it and click OK to move forword.**

***Problems encountered***

* How to contact with user efficiently.

The way I learned how to contact with the user is Scanner from the previous course. But when I trying to use it, I found out that it was to complicate to multiple contact with user. So, I was trying to search the better way in Google. I found out the new thing “gui” and discover it. I think the best way to do it is using JoptionPane.

* Error in the WebCrawler class

When I import the WebCrawler class, there are a branch of errors appeared. I was trying to revise this class a lot of times, but it seems like does not work. This problem takes me two days to solve it. The way to solve this problem is really easy, just downlow the jsoup class from internet and import it to the package. All the error in this class were disappear.

* Type of the set to store the link and the score.

The first time I did was used the ArrayList to store the links and the Array to store the score. The score could be sorted by using heapSort method but the links could not be sorted by ranking the score. And the other problem was the links and scores could not pair efficiently and automatically. So, the HashMap come into my mind and I trying to use it, which was convenient and efficient.

* Out of Bound.

I think the most times problem occurred is the Out of Bound. I converted the code of those seven methods from the power point and test them. I found that it always Out of Bound. That’s a tough thing and takes me a long time to solve it. I think the best way to solve this problem is differentiate the length and index between a heap and array. If I could and revise the method with the size issue, it could be overcome.

* Failed to extract max

The purpose of the extractMax method is to return the biggest value in the given array and the rest of the number still satisfy the max heap property. But I successfully return the biggest value but it still stay at the given array. I think I was misunderstanding the given pseudo code and the used of the extractMax. I think the requirement of the used of extractMax method is that the given array is satisfying the max heap property. What I did in the original project, the array I trying to call extractMax is sorted by heapsort method, which means it is ordered from the smaller to bigger. So the solution I found out is that adding the buildMaxHeap method at the beginning of extracMax method. And doing heapsort for the array after calling extractMax to make sure it get back to the sorted array.

* Disadvantage of HashMap

The HashMap could match value and key efficiently and it is convenient to get the key back by the value. But when we trying to get the key back by the value, it was tough. When I trying to use the scores to get the website, it would causes the list occurring duplicated website because some websites may have the same score.

***Lessons Learned***

First of all, I would like to say thank you so much to my professor, Mike Wu. He gave me the chance to finish this programming assignment and help me to learn the concepts and skills about the algorithm. From this programming assignment, I learned the concepts that what is the Google search engine and how it works. The Heapsort algorithm and Max-Heap Priority Queue did a great contribution for this programming assignment. I found out that how to use them in a real way and they are more efficient than I thought. The way to contact with the user I had learn is the Scanner, but it seems like only effective for single operation. As I search and discover the gui class. I knew how to use JoptionPane to contact with user. Actually, this is the first time I create my own application and write code in the midnight. I feel that the way to be a good programmer is not an easy path. I also learned that comment out for each method and major code is very important. When we write a lot of code, it was difficult to find back the code we want. We should write the comment for what is about the next code when we start to write the it. Debugging is the most tough thing in a programming project. If there is a way wrong, it takes a long time to debug to find it and solve it. Looking up the java API is also a good thing to programming, all the method are on it and it is easy to find them out. In this programming assignment, I also try to assume the user do not type in the thing or wrong thing the system wants, I make a lot of Exception for it and double check with the user. I think that is the good way to make a good application. We should not just test it in a programmer way but should test it in a user way. It is an amazing thing to see the result of the code we write. The process of running, debugging and revising is painful, but I really enjoy it. I know it is the only way to create a prefect program. No pain, no gain! Work hard, get the job!