**Programming Assignment 1**

**Simulator of Google Search**

**Engine Results Page (SERP) using Heap Sort**

CS 146 – Intro to Algorithm and Data Structure

Design by Linjun Cao

**Table of Contents**

1. Explain Google Search Engine Simulator design and implementation details --------------------------------- 3
2. Explanation of the Classes/Functions
3. Classes ---------------------------------------- 4
4. Functions --------------------------------------- 14
5. The procedure of how to unzip files, install application, and run/test codes --------------------------------------- 18
6. Problems encountered during the implementation ---- 19
7. Lessons Learned --------------------------------------- 20

**Part I: Explain Google Search Engine Simulator design and implementation details**

The Google Search Engine Simulator is based on a Web Crawler which collects the URLs from internet and an algorithm which processes and arranges this URLs. So, my program is divided by two parts, Web Crawler and Processing Program.

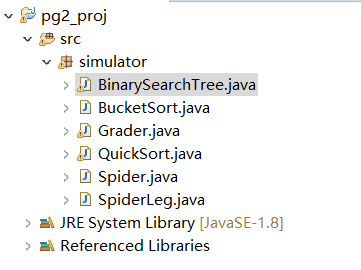
Since I didn’t study web crawler before, so I use open source code and make a little change. I choose Spider because it is straight forward, but it is enough for our demand.

The Processing Program is combined by Heap Sort algorithm and execute algorithm. The Heap Sort can sort 30 URLs by their score. Moreover, it can return the URL with greatest score by extract-max algorithm. The execute algorithm use random generator to give score to each URL and print out result by using the function above.

**Part II: 2. Explanation of the Classes/Functions**

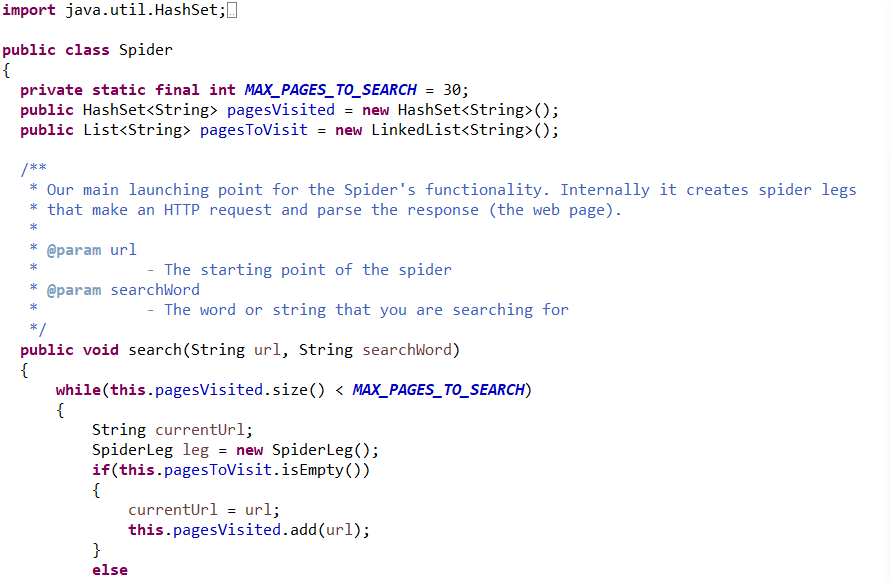
**Classes:**

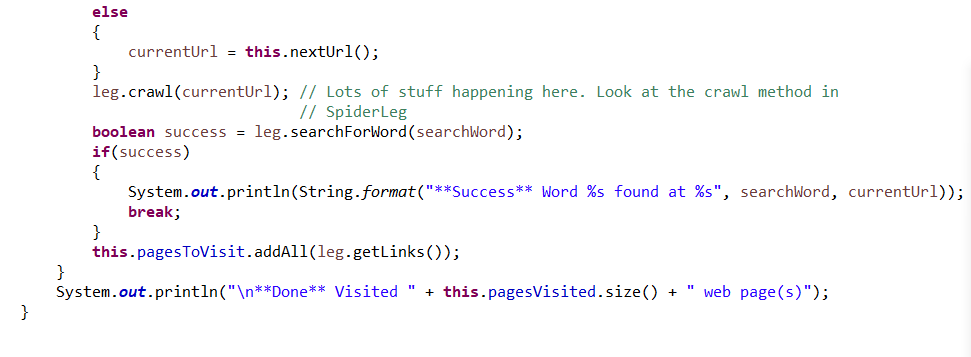
We have the following classes.

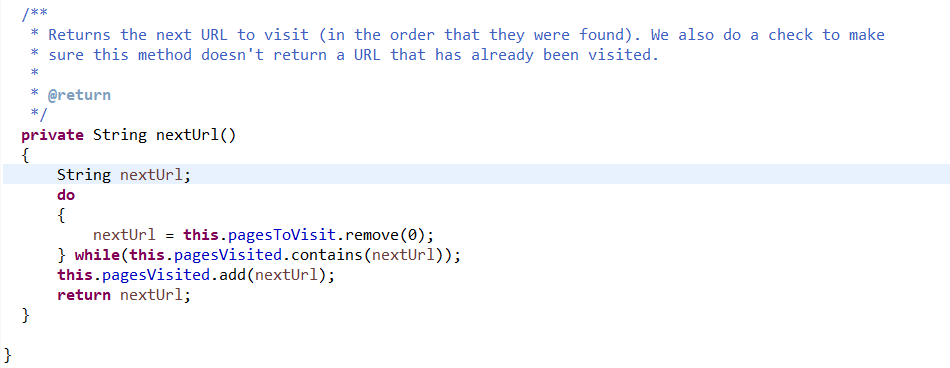


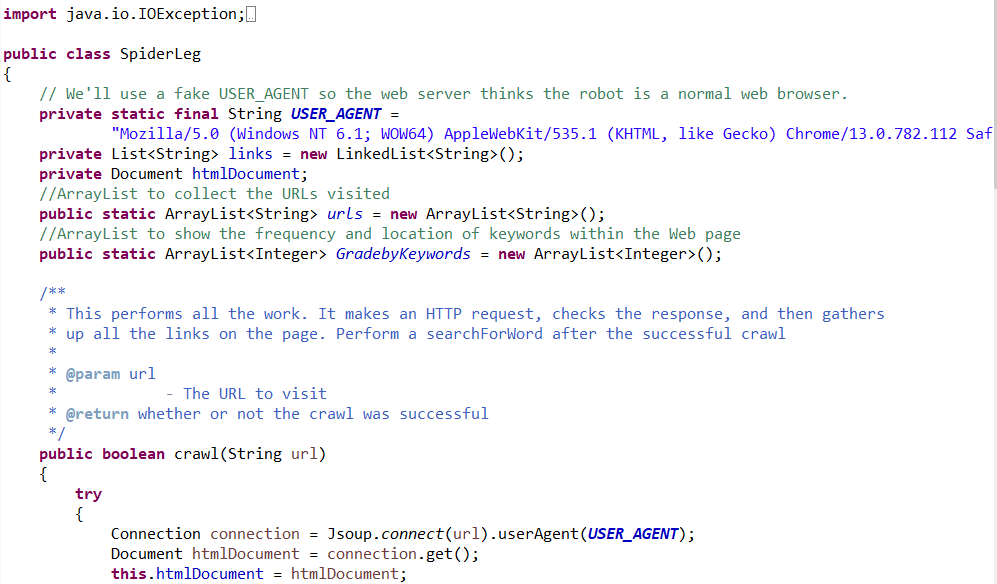
Class Spider and Class SpiderLeg are the part of web crawler. It ‘eats’ a link and a key word and give URLs which we need. An array is created to collect those URLs until it has already got 30 URLs.

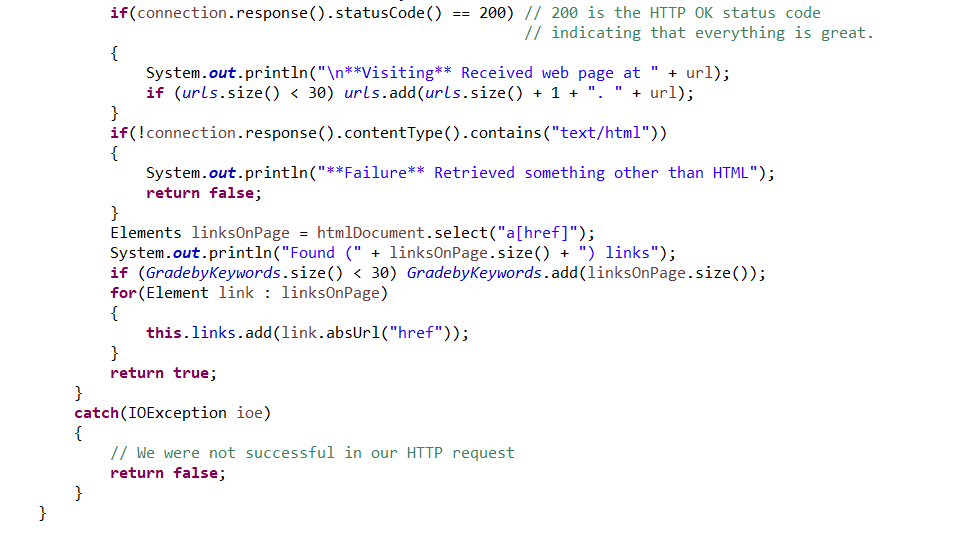
Here are some screen shots of these two classes:

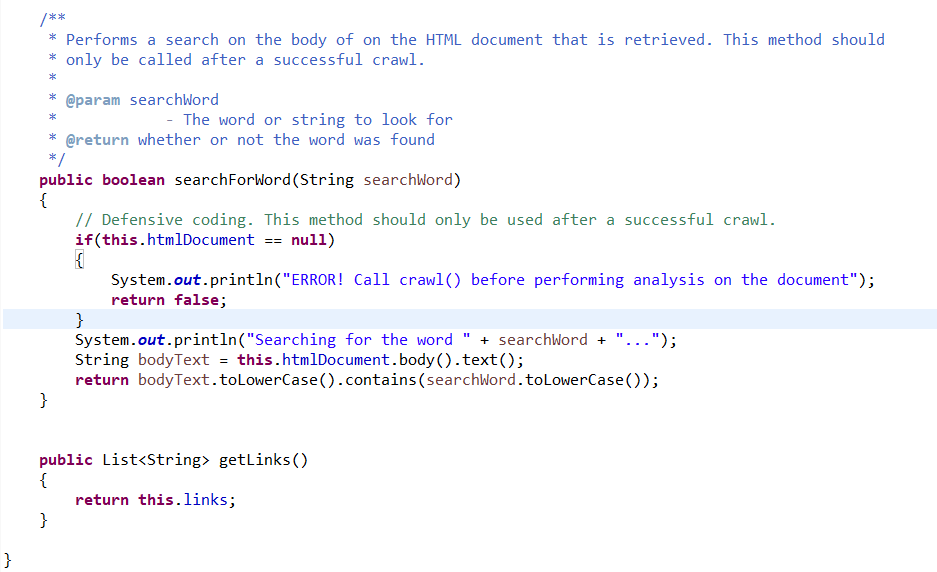




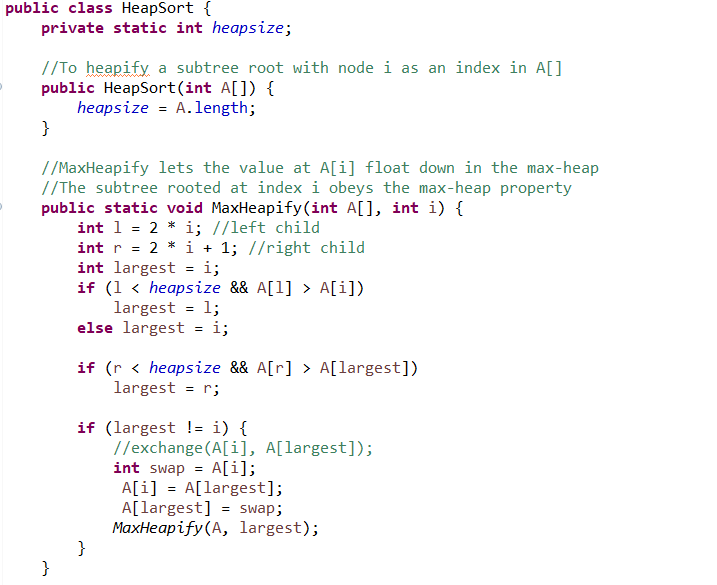


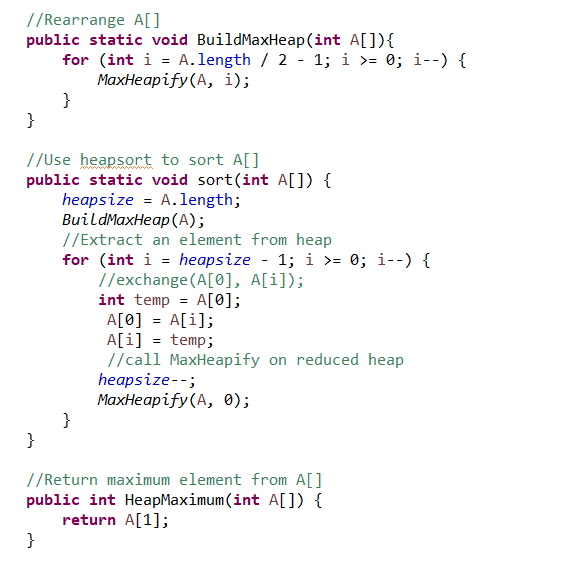


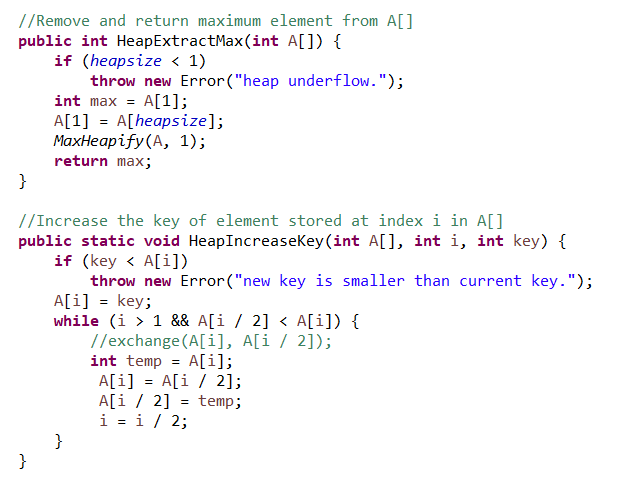


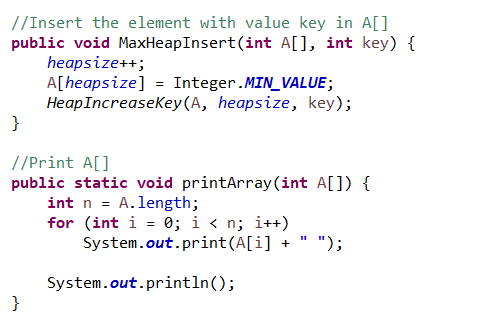


Class Heap Sort is the part to achieve heapsort. Here are some screen shots of it:

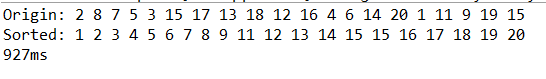




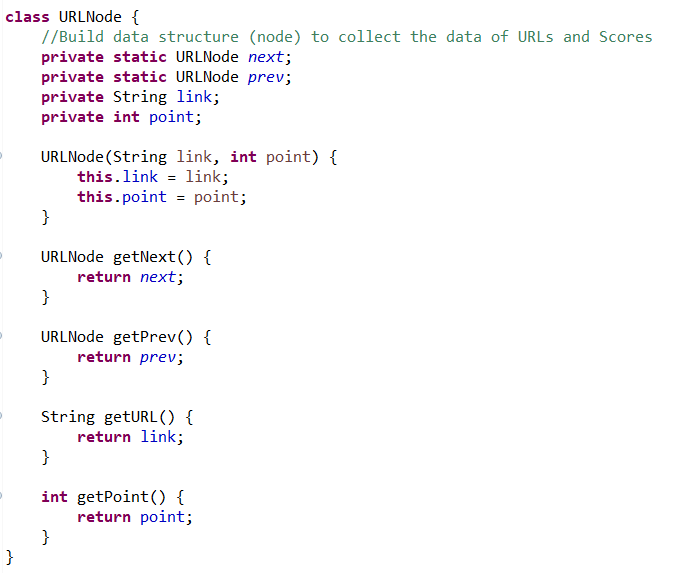




Here is a short test of quicksort:

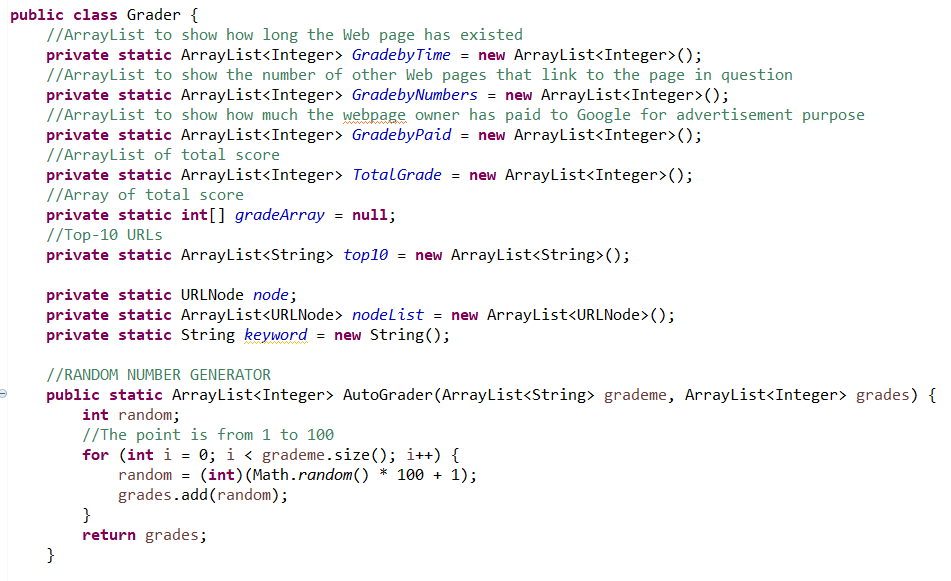


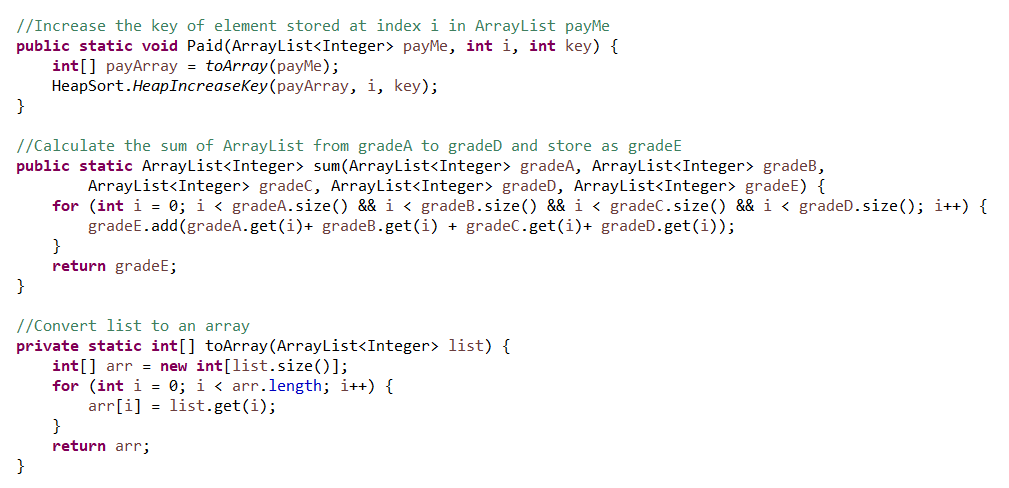
Class URLNode is the part to build Linked-List for the URL-Node. Here are some screen shots of it:

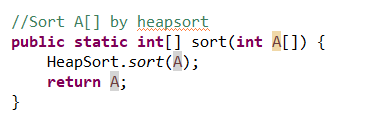


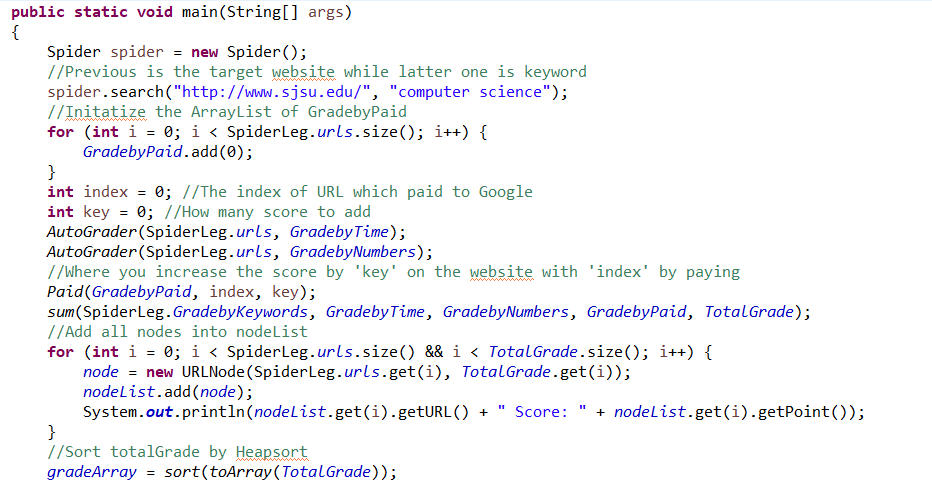
Class Grader is the executive class of the entire programming. In this class I collect 30 URLs from the web crawler, analyze and put them into the Linked-List. In the main method we can print all 30 URLs and their index and total score.

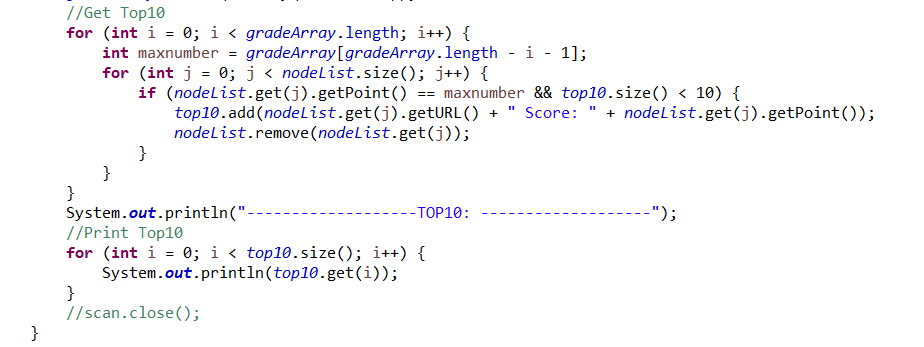
The screen shots of this class are the following:



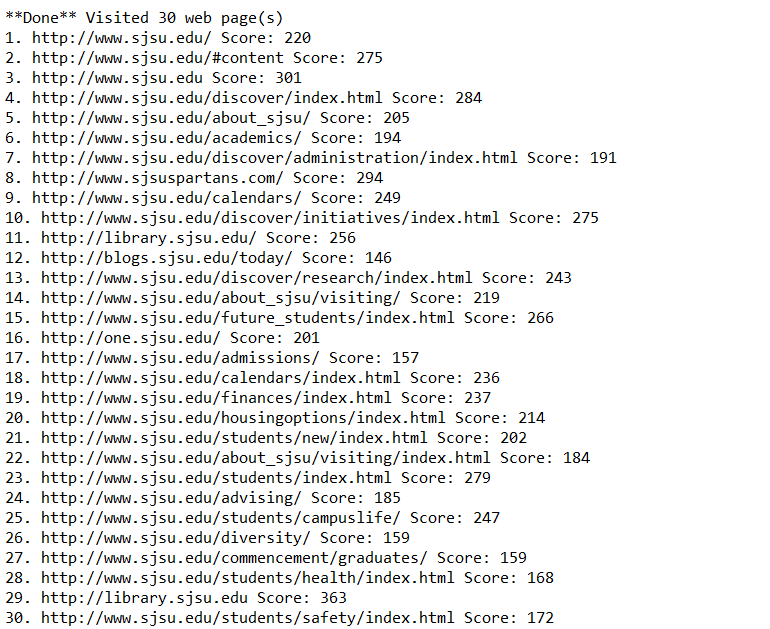




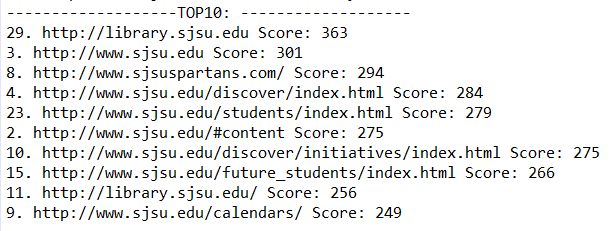




Here is the result of 30 URLs:



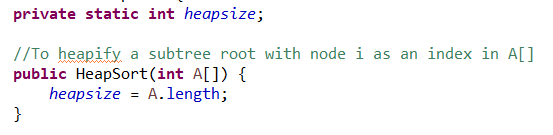
Here is the result of Top 10:



Functions:

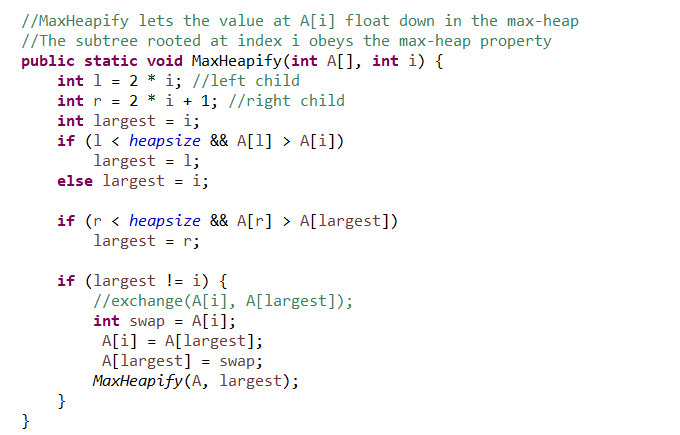
Class Heap Sort:

Constructor:

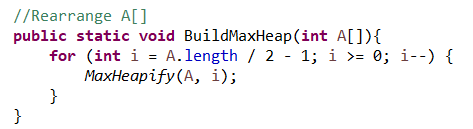


Initialize heap-size as the length of array.

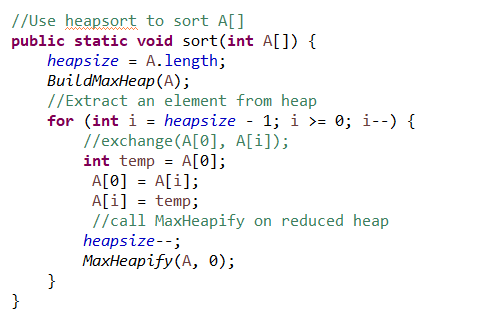
Function Max-Heap:



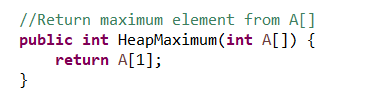
Function Build-Max-Heap:



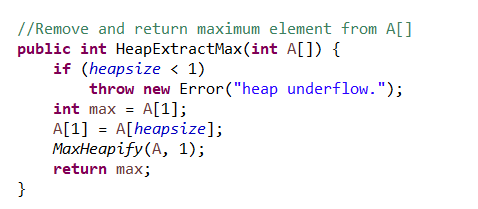
Function sort:



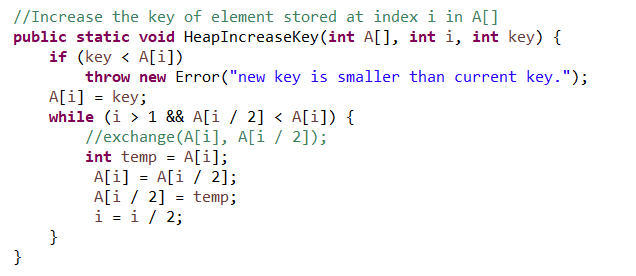
Function Heap-Max:



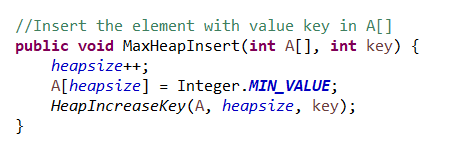
Function Heap-Extract-Max:



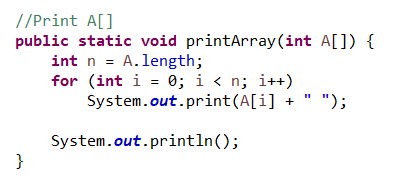
Function Heap-Increase-Key:



Function Max-Heap-Insert:



Function print-Array:

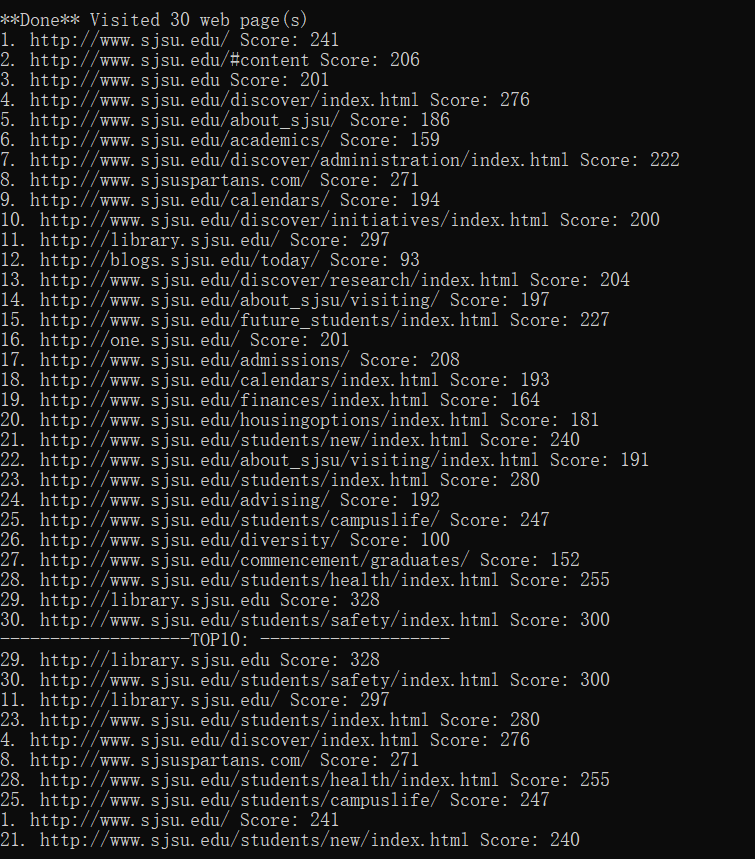


Part III. The procedure of how to unzip files, install application, and run/test codes:

First, unzip the zip and you will see a jar file called: Linjun-Cao\_PA1.jar. Then run the command and code in:

java -jar Linjun-Cao\_PA1.jar

And the jar file will run directly.



Part IV. Problems encountered during the implementation

I meet some problems when I implement Heap-Sort algorithm using the pseudo code from textbook. First is the heap-size. In the text book the define of heap-size is blurred. I didn’t know where to initialize it. Actually, it is initialized at very beginning as the length of the array, but it can be changed so it is different from the length of array.

Another problem I meet with is choose the data structure for the URL (String and total score). I use Hash Map at the beginning, but it doesn’t work well. Then I use linked list.

Part V. Lessons Learned：

I learn two things from this assignment. First, choose data structure carefully. There are many different types can be chosen. But not everyone is work.

Next, I can’t just enter the code in book into the program directly. It needs to be reversed by debugging.