Homework Assignment 1

(Programming Category)

Student Name: Jongyeon Kim
Student Session: CS4365

Design of crawler

It is really simple web crawler for College of Computing website which is ‘http://www.cc.gatech.edu/’. The purpose of this web crawler is getting related search word. This web crawler is designed by Java and I used Jsoup which can get HTML source code from website. When user type the search word, this web crawler get every subjects of searching result from web site by Jsoup. Finally, this web crawler can show a sorted list with words and count. This list shows that what word come out when users try to search word on College of Computing website.

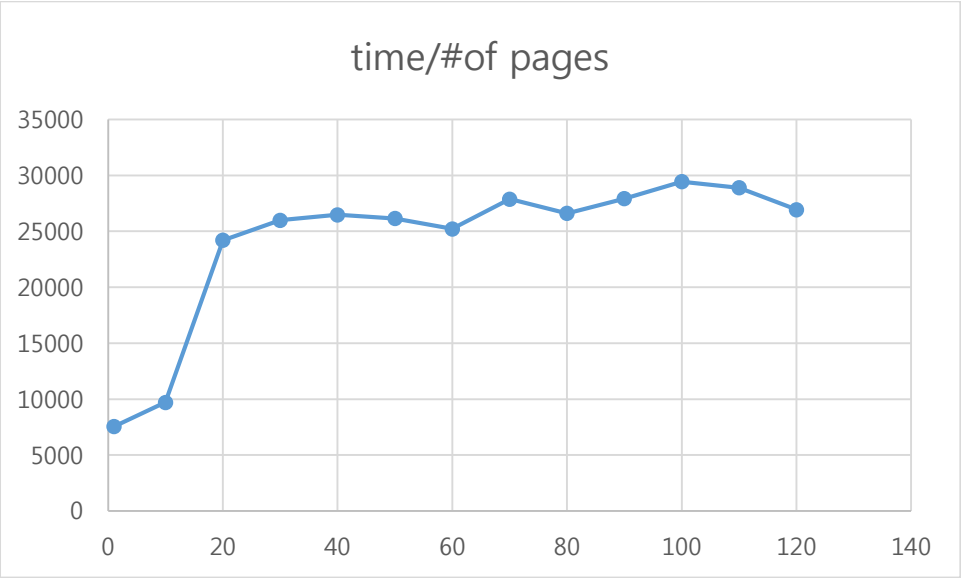
ScreenShots of Web Crawler

```
C:₩Users₩Administrator₩Desktop>java -classpath "C:₩Users₩Administrator₩Desktop₩j
soup-1.8.3.jar;." Crawler
What word you want to search? computing
How many page? 10

Result:

Computing=84
of=58
A=40
History=38
Hardware=37
retroTECH=37
in=18
College=11
and=10
Minor=10
Georgia=7
Tech=7
GT=7
to=5
Career=5
for=5
Research=5
on=5
at=4
Symposium=4
the=4
Program=4
Ubiquitous=4
Lecture=4
PhD=4
Fair=4
Series=4
Future=4
Beyond=3
Workshop=3
Sunbird=3
Awards=3
Cybersecurity=3
Focus=3
HumanCentered=3
Computings=3
Gift=3
```

From above result, we can see that when user searched the word ‘computing’, some words came out on the result many times such as ‘History’, ‘Hardware’ and ‘retroTech’. The Problem of this result is that many useless words are located at high rank such as ‘of’, ‘A’, or ‘in’.

Crawl Statistics

#of pages	time(ms)
1	7545
10	9704
20	24194
30	25991
40	26469
50	26139
60	25220
70	27877
80	26607
90	27924
100	29429
110	28887
120	26933



The overall graph shows that time is increasing slightly as User try to find more pages, but it is not linear because speed of this web crawler is really

depend on Internet speed since Jsoup should get HTML source code from website.

While I implemented this web crawler, I had chance to look many website's HTML source code, so it is really helped me refresh my knowledge of web computing.

Code

```
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;
import java.util.*;

public class Crawler {
	public static void main(String[] args)throws Exception{
		long startTime = System.currentTimeMillis();
		HashMap<String, Integer> hm = new HashMap<String, Integer>();
		//TreeMap<String, Integer> hm = new TreeMap<String, Integer>();

		Scanner scanner = new Scanner(System.in);
		System.out.print("What word you want to search? ");
		String inputst = scanner.next();
		System.out.print("How many page? ");
		int inputpg = scanner.nextInt();

		String word = inputst;
		int page = inputpg;

		for (int j = 0; j <= page; j++) {

			Document doc =
Jsoup.connect("http://www.cc.gatech.edu/search/node/" + word + "?page=" +
j).get();
			Elements titles = doc.select("div.no-sidebars ol li h3");


			StringTokenizer st = new StringTokenizer(titles.text());

			int count = 1;

			while(st.hasMoreTokens()) {

				String key = st.nextToken().replaceAll("[^A-Za-
z]","");


				if(hm.containsKey(key)) {
					count = hm.get(key) + 1;
				}
```

```
                        hm.put(key, count);
                        count = 1;
                    }
                }


                hm.remove("");

                ArrayList as = new ArrayList(hm.entrySet());

            Collections.sort(as, new Comparator() {
                public int compare(Object o1, Object o2)
                {
                    Map.Entry e1 = (Map.Entry)o1;
                    Map.Entry e2 = (Map.Entry)o2;
                    Integer first = (Integer)e1.getValue();
                    Integer second = (Integer)e2.getValue();
                    return second.compareTo(first);
                }
            });

            Iterator i = as.iterator();
            System.out.println("");
            System.out.println("Result:\n");
            while (i.hasNext())
            {
                System.out.println((Map.Entry)i.next());
            }

                scanner.close();
                long endTime = System.currentTimeMillis();
                System.out.println("");
                System.out.println("Took "+(endTime - startTime) + "
milliseconds");

            }
}
```