COSC 2320: Data Structures

Homework 6: Graphs and the World Wide Web

Deadline: **Dec. 6, 2014**

1. INTRODUCTION

In this homework you will create a C++ program to calculate page ranks in a small model of the World Wide Web, using graphs. A graph is defined as G = {V, E} where V is a set of vertices or nodes and E is a set of edges of links between the nodes.

In essence a webpage is a node in huge graph. All hyperlinks in the page are the edges in the graph, connecting it to other pages (other nodes). Evidently these edges have a direction, since "clicking" on a link takes you away from the page you were and unless there is a link back (assuming there is no "back" button in your browser) you couldn't return to where you were originally. This is called a "directed graph".

The degree of a vertex is the number of edges that are incident on the vertex. In a directed graph we can divide this number into two parts: the in-degree is the number of edges that are incident on the vertex and the out-degree is the number of edges that is "pointing outward" from the vertex. In this homework you will calculate the degree of each page of the test case we will provide. You should distinguish the in-degree and out-degree.

2. INPUT & OUTPUT

The input of the program is a file whose name is given in the command line argument. This input file will contain a list of webpage file names that you need to parse. Every webpage file name will take up a line. A webpage file will consist of random keywords and links to other webpage files. Those links will be clearly marked as in HTML language:

```
<a href = "test.html"> link </a>
```

We will strictly adhere to this convention. Furthermore since those files will have the ".html" extension you will be able to visualize them in your browsers. The output will consist of **three** parts separated by a blank line:

- The most referenced page (page that has the highest in-degree);
- A list of all broken links (links to files that don't exist);
- A list of all sinks or black holes (files with in-degree $\geqslant$ 0 but out-degree = 0).

We will make sure that:

- Each test case will have one and only one most referenced page.
- If the test case has no broken links, output "No broken links".
- If the test case has no sinks, output "No sinks".
- For broken pages and sinks, the output format should follow the example output provided in Section 4.

3. PROGRAM AND ARGUMENT SPECIFICATION

The main program should be called "**pagerank**". The program should be able to take the first argument as the input file.

The call syntax will be like:

```
pagerank.exe list.txt
```

Note that the file name will not necessarily be the same every time, so your program shouldn't have the input command file name hard coded.

4. A SAMPLE TEST CASE

"< EOF >" just signals the end of the file, it is not literally a string in the file.

list.txt:

```
page1.html
page2.html
page3.html
page4.html
page5.html
< EOF >
```

page1.html:

```
<a href = "page3.html"> page3 </a>
<a href = "page4.html"> page4 </a>
<a href = "page5.html"> page5 </a>
<a href = "page6.html"> page6 </a>
< EOF >
```

page2.html:

```
<a href = "page1.html"> page1 </a>
< EOF >
```

page3.html:

```
<a href = "page1.html"> page1 </a>
<a href = "page2.html"> page2 </a>
< EOF >
```

page4.html:

```
<a href = "page1.html"> page1 </a>
<a href = "page2.html"> page2 </a>
<a href = "page3.html"> page3 </a>
< EOF >
```

page5.html

```
< EOF >
```

Program call:

```
pagerank.exe list.txt
```

Output:

```
page1.html

page6.html Broken Link

page5.html Sink
< EOF >
```

5. SUBMISSION REQUIREMENTS

Your submission should be well tested before submitting under Visual Studio 2010 or later versions. You can get a copy of Visual Studio from the UH website using your cougarnet username and password. The URL is http://uh.edu/infotech/php/software/list.php.

We use the UH blackboard system to collect your homework submissions. Before you submit your homework, please make sure to **put everything in a ZIP file** named in the form of **LastName_PeopleSoftID_HW6.zip.**

For example: Zhang_1234567_HW6.zip

The instructions about how to use the blackboard system can be found on the TA's webpage for this course: http://www2.cs.uh.edu/~yzhang/cosc2320-f2014/

6. GRADING

The maximum grade for this homework is 100.

You will get 15 pts for submitting the homework in time, 10 pts if your program can be successfully compiled.

We will test your program with 5 easy test cases and 5 hard test cases. Each easy test case will worth 10 pts, and each of the hard ones will worth 5 pts.

When testing, we will compare your program's output with the standard output. Therefore **do not print any content on the screen unless required**, avoid any prompt information like "Please enter the input file name:", "The elements in the doubly linked list are:" etc.

Last but not least, **no cheating or plagiarism will be tolerated in any graded submissions.**