ECE 242 Project 3 Fall 2010
Searching and Sorting

Overview

In this project you will implement a program to find out the geographical distribution of the visitors to a web server. A web server periodically dumps a web log file that records the information (which includes the IP address) of every visitor. You will analyze a web log to identify the country of each visitor, given the mapping between IP addresses and countries. Then the number of visitors from each country can be calculated.

This service is widely available online for people to get a sense of who is interested in their website.

http://www3.clustrmaps.com/counter/maps.php?url=http://rio.ecs.umass.edu/#totals

This page shows the geographic distribution of the visitors to the website of the Multimedia Networking & Internet Lab. You will implement a program to generate a similar result as the one shown on this web page.

United States (US)	234
India (IN)	36
China (CN)	31
Thailand (TH)	14
Jordan (JO)	13
Iran, Islamic Republic of (IR)	4
France (FR)	4
Canada (CA)	3
Philippines (PH)	2
Pakistan (PK)	2
Ireland (IE)	2
Taiwan (TW)	2
Switzerland (CH)	2
Vietnam (VN)	2
Germany (DE)	2
Korea, Republic of (KR)	2
Ukraine (UA)	1
Belgium (BE)	1
Russian Federation (RU)	1
United Kingdom (GB)	1
Bahrain (BH)	1
Indonesia (ID)	1
Mauritius (MU)	1
Singapore (SG)	1
Myanmar (MM)	1
Brazil (BR)	1
Italy (IT)	1

In this project, you will be given a web log file, which contains a list of visitors and their information (more details coming soon). You will also be given a mapping file, which contains a list of IP addresses and their corresponding countries. The useful information in these 2 files needs to be extracted and stored in some data structures. Using these data structures, you will first find out the country of each visitor recorded in the web log. You will then calculate the number of visitors from each country and print out the result, which should be similar to the one shown on the right. You should also report the running time of each of your solutions.

Project description

These are the main steps involved in this project.

1. Parsing a web log file

A web log is a file automatically generated by a web server to log the information about the visitors. The following line is one entry in a web log.

38.101.148.126 - - [19/Sep/2010:06:35:16 -0400] "GET /robots.txt HTTP/1.1" 404 321 "-" "Mozilla/5.0 (compatible; discobot/1.1; +http://discoveryengine.com/discobot.html"

This log entry has the following information about this visitor.

Visitor's IP address, 38.101.148.126
Visit date, time and time zone, [19/Sep/2010:06:35:16 -0400]
HTTP command used, GET
Visited web page, /robots.txt

HTTP protocol, HTTP/1.1
Result status code, 404
Byte transferred, 321
User agent, Mozilla/5.0(compatible;discobot/1.1; +http://discoveryengine.com/discobot.html)

You will be given a real web server log file, "**web_log.txt**", which contains the recent visitors to a web sever. Each entry records a visitor to this website following the same format as shown above.

Although there is much information in each entry of the log file, in this project you will only use the visitor's IP address to find the location of the visitor. You will extract the IP address of each visitor from the web log file.

2. Find out a visitor's country

An IP address and country name mapping file "**ip_country_map.txt**" will be given to you for finding the correspondent country of a visitor. One entry in this file looks like this:

38.101 PL

This means any IP address with the **first** 16 bits (38 is the decimal form of a 8 bit number 00100110 and 101 is the decimal form of a 8 bit number 01100101) being 38.101 belongs to Poland (PL). This IP address and country mapping file is **NOT** sorted.

You will store all the information in this file to an array first. You will then implement **two separate approaches** to search this array and identify a visitor's country according to its IP address.

You will implement two ways of searching the IP address and country mapping array, the simple search and the fast search.

a) Simple search. Since the mapping file is not sorted, the simple search approach linearly searches the entire array to find out the correspondent country name of an IP address.

b) Fast search. The fast search approach first sorts the mapping array according to the IP address using **Merge Sort**, and then performs a **Binary Search** to find the country name of a specific IP address.

3. Print out the visitor countries and sort it by the number of visitors

You will implement a method to calculate the number of visitors from each country and store the result in a data structure.

The data structure then needs to be **sorted** according to the number of visitors. You will use **Insertion sort** to sort it. You will print out all visitor countries sorted by the correspondent number of visitors, similar to the one in the example shown earlier.

4. Expected results

Using the same web log file and the mapping file, you will analyze them and print out the **sorted** country names **two times**, one time with the simple search approach and another time with the fast search approach. Print out the running time of each of **the two passes**.

The main sorting and searching algorithms you need to implement in this project are:

1. Linear search, for searching the unsorted mapping array.
2. Merge sort, for sorting the IP address and country name mapping array.
3. Binary search, for searching the sorted mapping array.
4. Insertion sort, for sorting the visitor country names according to number of visitors.

Guidelines

The following is the suggested flow of your program.

```
Inputs: web_log_file, ip_country_mapping_file
Output: country_list

for (every entry in the web_log_file, get the ip_address)
{
  country_name = find_country_name(ip_address, ip_country_mapping_file);
  update_country_list(country_name);
}
sort_country_list();
print country_list;
```

The following are the classes and the basic methods you need to implement. You are welcome to add more for your own convenience.

1. Class WebLogData

a) **ReadLogData** - This method reads in each entry of the web log and extracts the IP address of each visitor.

2. Class MappingData

a) **ReadMappingData** - This method stores the IP address and country name mapping file to an array. This array will be the database you will search on.
b) **SimpleSearchData** – This method finds out the correspondent country of each visit by linearly searching the mapping array.
c) **SortData** – This method sorts the IP address and country mapping array according to the IP address. You must use the recursive version of **Merge Sort** to sort the mapping array. You are NOT allowed to use any of the standard sort methods available in the Java libraries!
d) **BinarySearchData** – This method goes over each entry in the web log. It then finds out the correspondent country of each visit by **Binary Search**. This is possible since the mapping array has been sorted.

3. Class CountryData

a) **UpdateCountryData** – The method updates the country name array when the country name of a visitor is identified. It counts the number of visitors from each country.
b) **SortCountryData** – This method sorts the country name array according to the number of visitors. You must use **Insertion Sort** to sort this array. You are NOT allowed to use any of the standard sort methods available in the Java libraries!

Hint: Since the web log file is big, you are encouraged to start with a small portion of the web log just to make sure your program works correctly.

Grading

You should submit your completed code by the date posted on the course website. You will receive 80 points for your code running accurately, and 20 points for the thoroughness and usefulness of your comments which should be included in the code.

1. Implementing class WebLogData (5 pts)
2. Implementing class MappingData (30 pts)
3. Implementing class CountryData (5 pts)
5. Parsing input files correctly (10 points)
6. Analyzing the web log correctly using **both** simple search and fast search (30 pts)
7. Commenting thoroughly and usefully (20 pts)