CS 2123
Summer 2014

Homework 7

Assigned: 7/28/2013
Due: 8/6/2013 11:59 pm

1. Hashing (30 pts)

Combine 7.4.1 and 7.4.2 into a single problem, writing a C program to read integers from a file (data.txt), use separate hash and rehash functions to search for each integer read from the file (i.e., each int is a key), and insert the int/key into the hash table if the search is unsuccessful.

For each search, output to the console how many rehashes were necessary to find or insert that particular key. Searches that are successful the first time (i.e., using the hash function) rehash 0 times. An example of desired output using h(k) = k % 1000 and rh(hr) = hr++ % 1000 could be:

```
SEARCH FOR 1101 REHASHES 0
SEARCH FOR 5101 REHASHES 1
SEARCH FOR 3101 REHASHES 2
SEARCH FOR 5101 REHASHES 1
SEARCH FOR 1102 REHASHES 0
etc,
```

NOTES:

a. use a statically allocated array of record structures between 150 and 300 elements. Don't use dynamic memory allocation for this.

b. you don't have to restrict your rehashing function to a linear probe (e.g., you can use double hashing or some other technique). But, it MUST be a rehashing function, not chain hashing.

c. you can omit the RECTYPE r field of the hash table structure, record, in the textbook.

The format of the data file is:

```
<int>
<int>
...
```

2. Understanding graphs (30 pts)

Exercise 8.1.1. in the textbook.

You only need to consider nodes A, B, C, D, and F. Exclude nodes E, G, and H from your answers (makes the matrices simpler).

3. Implementing a graph (40 pts)

Exercise 8.4.2. in the textbook.

This needs to be a complete C program. Your program should read a provided data file (data2.txt), build the graph, and determine if the graph is acyclic.
Hint: implement the depth-first traversal algorithm to determine this.

The format of the data file is:

```
<# of nodes>
<node label> <0,1> <0,1> <0,1>… <0,1>
<node label> <0,1>,<0,1>,<0,1>… <0,1>
…
```

where each line includes a label for that node and its entries in an adjacency matrix.
For example, a file containing:

```
3
A 0 1 0
B 0 0 1
C 1 0 0
```

has 3 nodes, A, B, and C. A has an edge to B, B has an edge to C, and C has an edge to A.

Deliverables:
Answers to homework problem 2 should be typed and submitted in a single document. Problems 1 and 3 should be submitted as separate C source code files. I need all of the C source that you produce for this assignment problems so that I may compile and execute it.

Archive and submit the files to Blackboard.