CSC 228-01 Data Structures and Algorithms, Fall 2019
Instructor: Dr. Natarajan Meghanathan

Project 5: Using a Hashtable to Determine the Common Elements in all the Lists

Due by: Oct. 24th, 11.59 PM

In this project, you will use a Hashtable to determine the common elements in all the lists. If an element appears more than once in one or more lists, the algorithm should capture the instances the element is present in all the lists.

You are given a code that implements a Hashtable as an array of linked lists. You are also given a main function that will create an array of Lists using the input variable values that you enter. Your task is to find the common elements in all the Lists and print them. Implement your algorithm as an extension of the code in the main function, as indicated in the file.

I suggest an idea (described below) to accomplish the above objective using a Hash table. You could use this idea or any other idea (that would use a Hash table) to determine the common elements in all the Lists. You should not use any additional space (barring a Hash table and a List to store the common elements across all the Lists) that would grow with the size of the Lists and the number of Lists.

A brief description of the suggested idea is as follows:

Create and maintain a List (called CommonElementsList that will store the common elements in all the lists that are scanned until then: this is a property that will be maintained throughout the execution of the algorithm). Setup the hash table (say, 'H') using the contents of the first List in the array of Lists as well as copy the contents of the first List to the CommonElementsList.

Now, go through a loop to scan the elements in the other Lists of the array of Lists.

Before beginning to scan a particular List, empty the contents of the CommonElementsList (you could set the nextNodePtr for the head node to be null to accomplish this).

For a particular List that is scanned

If a value in the List is in the Hash table H, then remove an instance of the value from H as well as insert the value to the CommonElementsList.

After scanning a particular List

Empty the contents of the Hash table H

Fill up the Hash table H using the contents of the CommonElementsList

Print out the contents of the CommonElementsList

// This will correspond to the common elements across all the Lists scanned until then

The contents of the CommonElementsList after scanning all the Lists will be the elements that are present in all the Lists. If a value appears more than once in all the Lists, then the number of instances of the value in the CommonElementsList should correspond to the number of instances the value is present in all the Lists.

Note: As part of your deliverables, you are required to develop a pseudo code for the above idea as well as analyze its time complexity and space complexity (without taking into account the space for the Hash table and the CommonElementsList). If you are using any other idea, you should provide a brief description of the idea along with a pseudo code as well as analyze its time complexity and space complexity (without taking into account the space for the Hash table and the CommonElementsList, if you use one).

A sample screenshot is shown below (note that there are at least three instances of '1' in all the five lists, so there are three instances of '1' in the CommonElementsList; similarly, there are at least two instances of '6' in all the five lists).

```
Enter the number of lists: 5
Enter the number of elements per list: 25
Enter the maximum value for an element: 10
Enter the size of the hash table: 7

Elements generated in List # 0
1 1 2 4 2 3 9 4 2 7 9 7 6 5 8 0 9 7 0 7 6 9 0 7 1
Elements generated in List # 1
3 9 0 9 9 3 8 4 9 5 8 1 3 7 4 6 3 6 6 5 2 1 7 1 0
Elements generated in List # 2
8 1 1 3 3 3 9 9 3 4 9 1 6 8 1 4 0 6 2 8 0 1 6 1 0
Elements generated in List # 3
6 6 4 9 6 1 1 2 3 4 6 1 9 4 8 5 1 5 6 0 7 5 1 8 7
Elements generated in List # 4
5 1 1 2 0 8 1 5 6 8 5 6 6 8 1 9 5 2 7 6 8 7 2 7 5

Common elements from list # 0 to list # 1
3 9 0 9 9 8 4 9 5 1 7 4 6 6 2 1 7 1 0

Common elements from list # 0 to list # 2
8 1 1 3 9 9 4 9 1 6 4 0 6 2 0

Common elements from list # 0 to list # 3
6 6 4 9 1 1 2 3 4 1 9 8 0

Common elements from list # 0 to list # 4
1 1 2 0 8 1 6 6 9

Common elements in all the lists
1 1 2 0 8 1 6 6 9
Press any key to continue . . .
```

Submission

1 - 75 pts) A .cpp file containing the entire code, including the main function featuring your implementation of the algorithm to determine and print the common elements in all the lists

2) A single PDF file featuring the following:

- a - 10 pts) Pseudo code of the algorithm to determine the common elements among Lists
- b - 5 pts) Analysis of the time complexity of the algorithm. Assume any lookup/read operation in a Hashtable can be executed in $\Theta(1)$.
- c - 5 pts) Analysis of the space complexity of algorithm. Do not consider the Hashtable and the CommonElementsList as part of the space usage.
- d - 5 pts) A screenshot of the execution of the code and displaying the outputs as described (similar to a format shown in the sample screenshot).