EEM 480 Algorithms and Complexity HW 4

In this Homework you are required to design a Facebook-like environment and program using hash structure. Your program will work on command line where the commands are given in one line as:

I <Name> - Creates a person with the name given in the line. If <Name> has been created before, print “<Name> can not be created”.

F <Name1> <Name2> - Sets the two people where the names given in the line are friends. If either <Name1> or <Name2> is not a created before print “<Name1> <Name2> can not be friend”.

E <Name1> <Name2> - Erases the friendship (<Name1> and <Name2> are no more friends). Print <Name1> and <Name2> are no more friends. If <Name1> <Name2> is not a friend before print “<Name1> <Name2> can not be erased”.

D <Name1> - Deletes the record of <Name1> from the structure and all the friendships related with <Name1>. If the <Name1> is not found, it prints “<Name1> is not in the list”.

L <Name> - Lists the friends of the person <Name>. If <Name> has no friend, print “<Name> has No friend”. If the <Name> is not found it prints “<Name> is not in the list”.

S <Name1> <Name2> - Checks whether the two person are friends or not. If so print “Yes”, If not print “No”.

O <Name1> - Prints the friends <Name2> , <Name3> , … of <Name1> and the friends of <Name2> and <Name3> … by inserting “*” in front of them. If the <Name1> is not found, it prints “<Name1> is not in the list”. If <Name1> has no friend, print “<Name1> has No friend”

R </DIR/Filename.txt> - Opens a file with the given name. In this file, it is supposed that each line contains a command of the program as given in Figure 1:.

X – Exit the program.

In the Program :

- The commands “I”, “F” and “S” will be executed in O(1) complexity which is independent of the total number of people in the system.
- You cannot use Java’s libraries. You have to implement every details of data structure you use.
- The command “L”, “E”, ”D” and “O” have to be executed in number of friends complexity = O(n)

Data Structure:

1. Define a Person class as given below

```
public class Person {
    String name;
    Person next;
}
```

2. The friends will be stored in the linked list using Person class.
3. If <Name1> and <Name2> are friends S <Name1> <Name2> and S <Name2> <Name1> will produce "Yes".
4. You have to store each person in a hash table. In order to resolve the collusion, propose and implement your own collusion resolving method. Clearly explain those methods in your report.
5. Propose a method to check S <Name1> <Name2> in O(1) complexity. (Hint : Don't check the friends linked list. Because it will be in O(n) complexity).

```
I Ali
I Veli
I Lutfullah
I Cevziye
F Cevziye Lutfullah
F Cevziye Ali
L Veli
L Cevziye
L Lutfullah
S Ali Veli
S Lutfullah Cevziye
L Cezviye
O Ali
D Cezviye
L Ali
E Ali Veli
```

Figure 1. Input.txt file

The Output :

```
Veli has No friend

Lutfullah Ali
```

```
Cevziye

No

Yes

Cezviye is not in the list

Cevziye

*Lutfullah

Cezviye is deleted

Ali has No friend

Ali Veli can not be erased
```

Rules for HW Submission

- You have to write your HW in NetBeans environment.
- You have to write a report with name "Report_HW4.pdf" explaining your HW (purpose, how did you solve it, what complexity you have, etc.) and what environment you used (NetBeans, for example). The person who read your report can easily use the class you have written.
- Discuss the result you have obtained.
- Submission should be in the form of a zip. When extracted, the result should be a single folder with the name "HW4".
- Don't forget to put your report into the zip file.
- The name of your project will be "**Name_Surame_HW4**. e.g. *Lutfullah_Arici_HW4*. If you do not obey the rule I will not grade your homework.
- You have to bundle your whole project folder into your HW4.zip file.
- If I extract your project file, then import to my environment and if it doesn't work, you will be graded on 30 not 100. (Double check. It saves life)
- Do HW by yourself. Be honest.