Lab 4 : DS II

Perfect Hashing

&

Universal Hashing

Members

Louay Magdy AbdelHalim	19016195
Mohamed Ayman Said	19016250

Github Repo.: click here

Program Structure

The program is composed of a main package called ***IHashing*** including :

- A package for Static Perfect Hashing

  It includes:

  - An interface
  - A class for O(N) space soln.
  - A class for O(N^2) space soln.
- A package for dynamic PerfectHashing

  It includes:

  - An interface
  - A class for O(N) space soln.
  - A class for O(N^2) space soln.
- *Item* class containing a key and a generic value
- Universal HashFn generator class

  It randomises a hash function according to input set size using matrix method and returns the no of times it takes to rehash
- Main class

Design Patterns Applied

- Interface Design pattern
- Factory Design Pattern

More details will be found in Algorithms Section

Comparison between O(N) space method and O(N ^2) space method in terms of **no of times of building the hash function on** …in Static Perfect Hashing

Click here

Algorithms

Click here

Sample Runs

1.

```
==============================
Select one of these types:
 1. for Static Perfect Hashing O(N square) space method.
 2. for Static Perfect Hashing O(N) space method.
 3. for Dynamic Perfect Hashing O(N square) space method....extra.
 4. for exit.
=> 2
enter a list to read in the following format:
{<k1, val1>, <k2, val2>, <.., ..>, <.., ..>}
....do not press enter till you finish ; so that all input is read
=> {<584, -183123396>, <235, 1845285240>}
Collision at index 1 with size of 2
no. of rehashings : 2
sizeOccupied = 4
select an option :
 1. look for a key
 2. exit
=> 1
enter the key: 235
value: 1845285240
```

2.

```
Perfect And Universal Hashing
=============================
Select one of these types:
 1. for Static Perfect Hashing O(N square) space method.
 2. for Static Perfect Hashing O(N) space method.
 3. for Dynamic Perfect Hashing O(N square) space method....extra.
 4. for exit.
=> 1
{<510, 454033791>, <184, 2002722154>, }
no. of rehashings : 1
sizeOccupied = 4
select an option :
 1. look for a key
 2. exit
=> 1
enter the key: 100
key not found!!
```

3.

```
Select one of these types:
 1. for Static Perfect Hashing O(N square) space method.
 2. for Static Perfect Hashing O(N) space method.
 3. for Dynamic Perfect Hashing O(N square) space method....extra.
 4. for exit.
=> 2
enter a list to read in the following format:
{<k1, val1>, <k2, val2>, <.., ..>, <.., ..>}
....do not press enter till you finish ; so that all input is read
=> {<982, 1846454005>, <389, -734363618>, }
Collision at index 0 with size of 2
no. of rehashings : 2
sizeOccupied = 4
select an option :
 1. look for a key
 2. exit
=> 1
enter the key: 100
key not found!!
```