BLM3021 Algorithm Analysis
Assignment - II

Group 2

Instructor: Assist. Prof. M. Amaç GÜVENSAN

Mounes Zawal – 18011906

Screen shots:

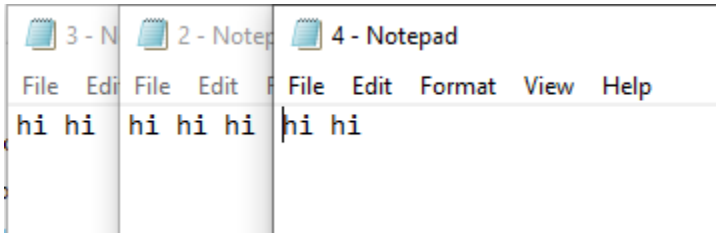
Trying to insert three files including the same word,

Expectations:

If we run the program, after inserting the files the table Size should be just one;

When searching a word, the files will be:

2.txt 3.txt 4.txt



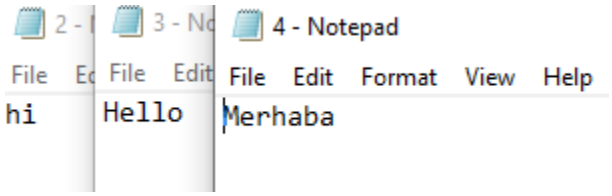
LET'S Try:

```
-----------------
(1) --> Insert New File.
(2) --> Look up a Word.
(3) --> Print Dictionary.
(4) --> Clear screen.
(999) --> Exit.
Please Enter an option :2
Enter the Word : hi
HI is found in these documents:
2.txt   3.txt   4.txt
-----------------
(1) --> Insert New File.
(2) --> Look up a Word.
(3) --> Print Dictionary.
(4) --> Clear screen.
(999) --> Exit.
Please Enter an option :3
{
        "Index":"311",
        "Word":" HI",
        "Locations":"2.txt","3.txt","4.txt"
}
Press any key to continue the program:
```

```
Table size: 1
1 Used from 997
Load factor: 0.001003
```

```
(1) --> Insert New File.
(2) --> Look up a Word.
(3) --> Print Dictionary.
(4) --> Clear screen.
(999) --> Exit.
Please Enter an option :2
Enter the Word : hello
This word is not found in the dictionary
```

IT WORKS!



For this case:

Table size would be: 3

```
----------------
(1) --> Insert New File.
(2) --> Look up a Word.
(3) --> Print Dictionary.
(4) --> Clear screen.
(999) --> Exit.
Please Enter an option :3
{
        "Index":"55",
        "Word":"HELLO",
        "Locations":"3.txt"
}
{
        "Index":"311",
        "Word":" HI",
        "Locations":"2.txt"
}
{
        "Index":"929",
        "Word":"MERHABA",
        "Locations":"4.txt"
}
```

```
Table size: 3
3 Used from 997
Load factor: 0.003009
```

```
(1) --> Insert New File.
(2) --> Look up a Word.
(3) --> Print Dictionary.
(4) --> Clear screen.
(999) --> Exit.
Please Enter an option :2
Enter the Word : moshi
This word is not found in the dictionary
-----------------
```

Using more then 80% of the table:

```
WORNING:
You have used 0.80 of the table Size
                --> Your last insertions are loaded into the memory
                --> Please Press any key to start the program
```

Now we will try to insert larger files, and have a look at the resulting dictionary.txt:

Inserting: A.txt, B.txt, C.txt, 1.txt

```
Table size: 673
673 Used from 997
Load factor: 0.675025
```

Dictionary - Notepad

File Edit Format View Help

```
{
        Index:1,
        Word:OCCUR,
        Locations:1.txt
}
{
        Index:2,
        Word:END,
        Locations:A.txt,B.txt
}
{
        Index:4,
        Word:RETURNED,
        Locations:A.txt
}
{
        Index:5,
        Word:ROOTS,
        Locations:1.txt
}
```

Dictionary - Notepad

File Edit Format View Help

```
{
        Index:318,
        Word:MAN,
        Locations:1.txt
}
{
        Index:319,
        Word:POTENTIAL,
        Locations:A.txt,B.txt,C.txt
}
{
        Index:321,
        Word:RECENTLY,
        Locations:1.txt
}
{
        Index:322,
        Word:MAGNAM,
        Locations:1.txt
}
```

Analysis:

Adding file's words:

$$f(n) = f_{insert}(n) + f_{search}(n)$$

best case:

if the word found in the table without collision existence:

then we will add the file's name to the locations:

this will take:

$f_{best}(n) = f_{search}(n) \in \Theta(m)$ ***where m: number of files existed***

Of course, if the word is found there will be no insertion:

Worst case:

When the world is not found, then we insert it:

$f_{search(worst)}(n) \in \Theta(M)$ ***where M: Table Size***

$f_{insert(worst)}(n) \in \Theta(M)$ ***where M: Table Size***

/when the table is almost full/

$f_{worst}(n) \in \Theta(M)$

Looking up a word:

$f_{search(best)}(n) \in \Theta(1)$

// if word is found from the first iteration without collisions

$f_{search(worst)}(n) \in \Theta(M)$

//when it iterates over the entire table