FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

Tinaztepe Campus, Buca-Kaynaklar, Dokuz Eylul University, IZMIR, TURKEY

Project Final Report: Simple Search Engine Date: 08.12.2022

Prepared By: Güney Söğüt

1- **Introduction:** The purpose of this report is to give information about the Simple Search Engine project.

2- **Project Description:** The aim of the project is to develop a text-based simple search engine. The program asks user to enter 3 words to search and then the program gives the most relevant document in the database.

3- **Algorithms:**

How it works? : Firstly, the program finds the stop words from stopwords.txt file. Then, detects the article files and split the words in the .txt files by DELIMITERS. After these steps, it checks whether the word is stop word or not. If not, the word will be added to Hashed Dictionary. After adding step, the program takes an input from the user and try to find the most relevant document.

Add function: It has 4 parameters as follows: key(word), value(file name),object array(file names + " " + word counts) and Boolean(check).
Firstly, it checks whether the load factor is passed or not. If yes, it rehashes. Then the method calls the indexing function (SSF or PAF). Then, if the found index on the table is null there is a new table entry, if not null, it means that the new key exists in the table before. Therefore, it increases the word count in the relevant file on the object array. Then, set the new array by updated.

increaseFileCount method: It takes the count array and searched object then it checks the file number then update the count value of that file in the array.

How to find most relevant document? :

For word 1: occurrence/all word counts in the current file

Apply this to the whole 3 inputs. Then, sum the outputs and check for the most relevant(bigger is more relevant).

4- Performance Table:

Load Factor	Hash Function	Collision Handling	Collision Count	Indexing Time	Avg. Search Time	Min. Search Time	Max. Search Time
α=50%	SSF	LP	3130379	207654600ns	42725ns	27000ns	330100ns
		DH	360818	70290500ns	11996ns	1900ns	197000ns
	PAF	LP	665	32348800ns	403ns	200ns	7200ns
		DH	1138	30000000ns	652ns	700ns	7600ns
α=80%	SSF	LP	3130379	186445500ns	40736ns	27000ns	267900ns
		DH	360818	67975800ns	11507ns	1900ns	152200ns
	PAF	LP	5710	36396300ns	708ns	300ns	9100ns
		DH	9244	34702600ns	783ns	300ns	21300ns

The most efficient way is PAF + DH with load factor of %50
The worst one is SSF + LP with load factor of %50