DESIGN AND IMPLEMENT A WORD CHECKER ALGORITHM

h

WORD CHECKER ALGORITHM:

- The word checker algorithm compares every word you type to its own list of words in the language you are using.
- Every time you type a string of letters and end it with one of these things, the word checker compares that string to a list of words that has been programmed into it. If your string of letters is not in the utility's internal list, it flags it.
- word checker would only have to load up a huge array of strings with words from a file on disk - then split the incoming text into words - and search for them on the list using a case-insensitive comparison function.

Time Complexity Analysis For Binary Search Tree and Hash Table For Searching A Word In A File

BINARY SEARCH TREE

- There are two types of tree data structure one is binary tree another is binary search tree.
- Searching in a binary search tree is more efficient than the binary tree.
- While inserting we will sort the data using less than and greater than operators
- So , we don't need to sort the data again.
- This is binary search tree usage.

HASH TABLE

- Hash table is a versatile data structure which is mathematically accurate and fast than most of the other data structures
- This data structure consists of key , value pairs.
- The value is identified by the key and the value is passed through a hash function and is stored as a hash
- Every element is unique in hash table
- Very efficient is searching inserting etc..

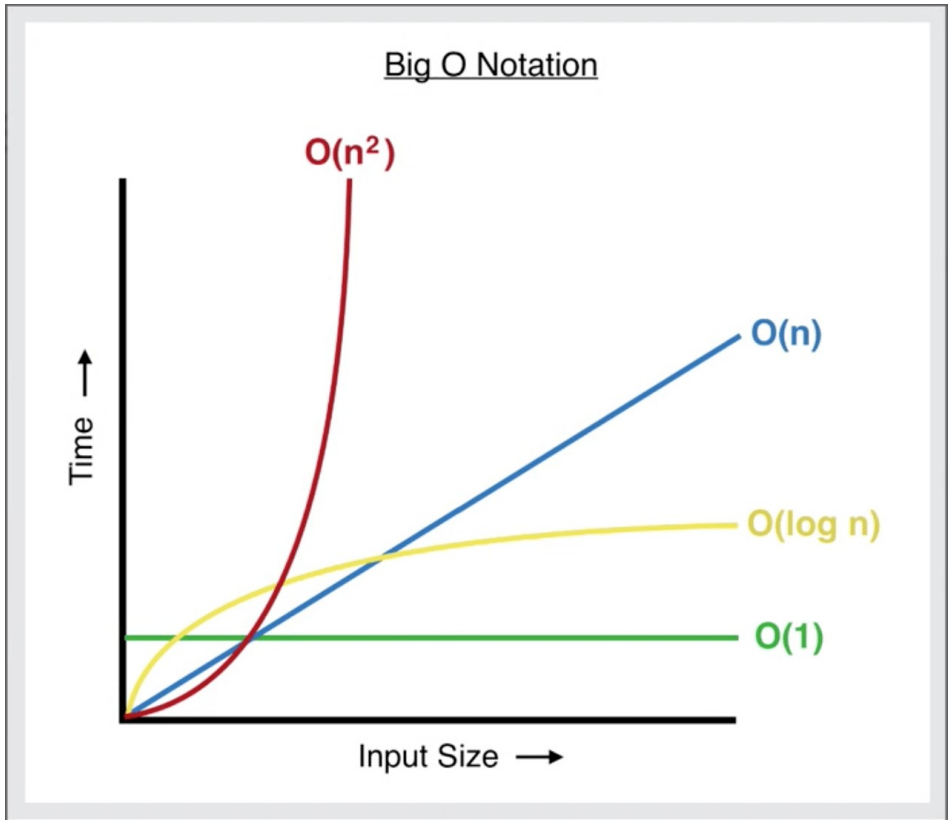
WHAT IS TIME COMPLEXITY?

- Time complexity is the calculation of how fast an algorithm at runtime.(when executing the program)
- In Computer science it is very common to test the time and space complexity of an algorithm
- The speed of an algorithm depends upon the organization of the data and how efficiently the data is sorted and how the data is fetched, inserted , deleted.
- There are many data structures such as array, linked list , tree , graph, hash table etc…
- For our purpose of checking the word is present or not in the given file we are considering Binary search tree and hash table.

BIG (O) NOTATION

- Big (O) refers to the analysis of complexity of algorithm irrespective of the hardware the algorithm is running on
- O(1) means constant time access for inserting, deleting , retrieving
- O(n) means it depends on `n` number of elements in it
- O(log n) is logarithmic time
- O(n^2) is square of number of elements in it

TIME COMPLEXITY COMPARISON



CONCLUSION

- Based on our experiments,
- Hash Map is O(1) time complexity and BST has O(log n) time complexity.
- And O(log n) is always better for any algorithm
- So , Comparing the two data structures we think that the Binary Search Tree is better for our purpose of word searching.

OUTPUT:

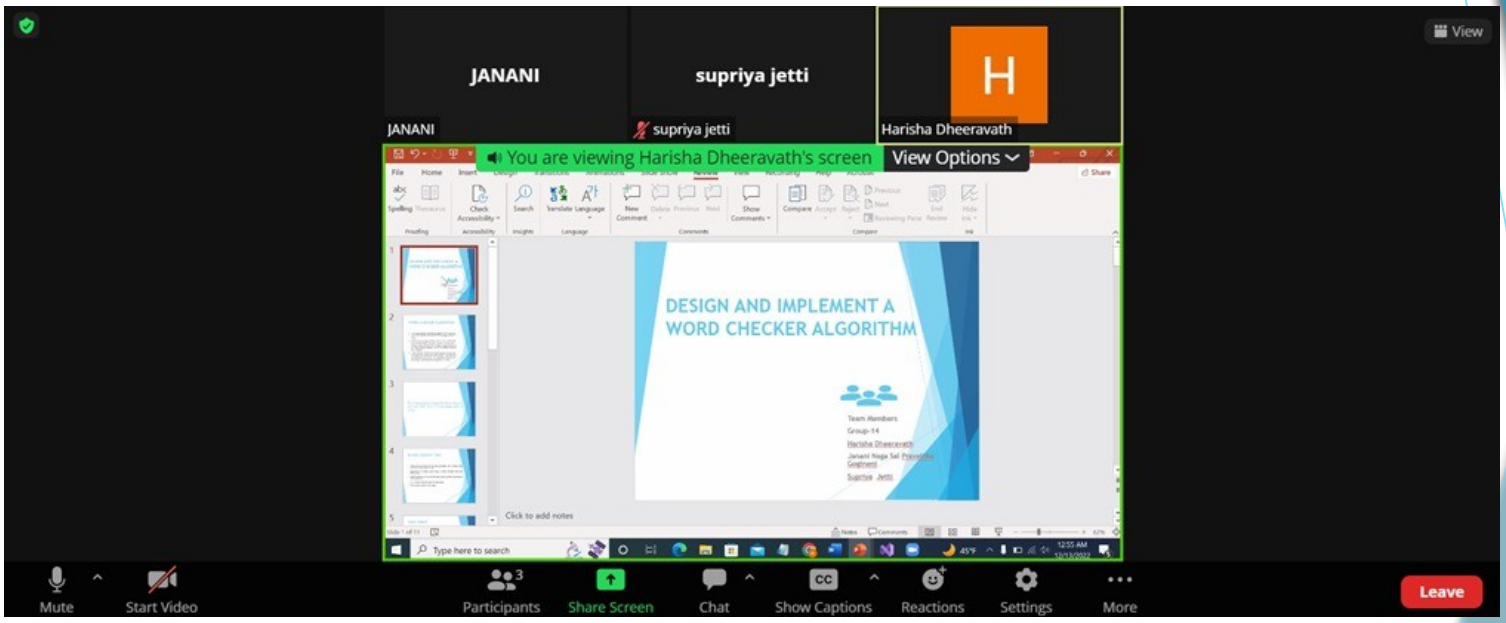
```
124             else
125             {
126                 cout << "Word not found" << endl;
127             }
128             endTime = chrono::high_resolution_clock::now();
129         }
130         else
```

```
Developer PowerShell
Developer PowerShell
Word found
Time Taken : 502 microseconds
PS C:\Users\Student\Desktop\Word Checker> .\main.exe
Enter the file name:
100000.txt
Enter the word name to search:
wobqtp
Select which data structure you want to use:
1. Binary Search Tree
2. Hash Table
1
Word found
Time Taken : 185 microseconds
PS C:\Users\Student\Desktop\Word Checker>
```

```
118             // if the word is found in the table
119             startTime = chrono::high_resolution_clock::now();
120             if (table.find(word) != table.end())
121             {
122                 cout << "Word found" << endl;
123             }
124             else
125             {
126                 cout << "Word not found" << endl;
127             }
128             endTime = chrono::high_resolution_clock::now();
129         }
130         else
```

```
Developer PowerShell
Developer PowerShell
Word found
Time Taken : 365 microseconds
PS C:\Users\Student\Desktop\Word Checker> .\main.exe
Enter the file name:
100000.txt
Enter the word name to search:
vtouqn
Select which data structure you want to use:
1. Binary Search Tree
2. Hash Table
2
Word found
Time Taken : 368 microseconds
PS C:\Users\Student\Desktop\Word Checker>
```

GROUP IMAGES DISSCUSSION:



THANK YOU