**FREQUENCY COMPUTATION OF A STRING IN A TEXT USING**

**HASH TABLE, BINARY SEARCH TREE AND REDBLACK TREE**

**Description:** Frequency computation of a string in a text using data structures hash table, binary search tree and redblack tree.

**Purpose:** The proposed package has a goal of computing the number of occurrences of a string in a given text. The computation is carried in three different ways employing the concepts of hash table, binary search tree and red black tree. The implementation with respect to each data structure is written in independent modules. These modules shall be tested with data to verify functionality.

**Input Data :** A text file containing strings or manual input by the user

**Language Used :** Python

**Data Structures Used:**  Binary Search Tree, Red Black Tree, Hash Table

**The organization of package code in the repository:**

1.     Main\_Module.py :  Takes input from the user to display the output of chosen functionalities.

2.     BST\_module.py :  Contains methods for insertion, deletion , Inorder Traversal, Searching, successor, predecessor , Minimum and Maximum keys

3.     RBT\_module.py :  Includes the methods for Insertion, deletion, Searching, Successor, Predecessor, Inorder Traversal and logic to increment the value when inserted duplicate keys

4.     HT\_module.py : Functionalities for Insertion, deletion, search, Listing the keys and values are implemented.