UNIT II

ASSIGNMENT- PART I

Q.1] A Dictionary stores keywords & its meanings. Write a program for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword.

Q.2] Implement Dictionary ADT in two ways for the word-line concordance application.

1. Using Binary Search Tree
2. Using AVL Tree

Also further compute the execution time (in seconds) taken by both dictionary ADT implementations.

Required Files: 1. The textual information to be examined is in the file: "data.txt"

2. The stop words are in the file "stop_words.txt"

[Hint:

1. The word-concordance application code and each dictionary ADT should be developed and tested separately.
2. A linked list implementation of the queue ADT can be useful to store the lines-of-occurrence for each word.
3. use a regular expression to define a word can be studied at: https://docs.python.org/dev/howto/regex.html

]

The goal of this assignment is to process a textual, data file (data.txt) to generate a word concordance with line numbers for each main word. A dictionary ADT is perfect to store the word concordance with the word being the dictionary key and a list of its line numbers being the associated value with the key. Since the concordance should only keep track of the "main" words, there will actually be a second stop-words file (stop_words.txt). The stop-words file will contain a list of stop words (e.g., "a", "the", etc.) -- these words will not be included in the concordance even if they do appear in the data file. The output of the program should be a text file containing the concordance words printed out in alphabetical order along with their corresponding line numbers. Sample files might be:

Sample “stop_words.txt”

```
a
about
be
by
can
do
i
in
is
it
of
on
the
this
to
was
```

Sample “data.txt” file

```
This is a sample data (text) file to
be processed by your word-concordance program.

The real data file is much bigger.
```

Sample output file

```
bigger: 4
concordance: 2
data: 1 4
file: 1 4
much: 4
processed: 2
program: 2
real: 4
sample: 1
text: 1
word: 2
your: 2
```

Notes:

1) Words are defined to be sequences of letters delimited by any non-letter. (e.g., white space, punctuation, parentheses, dashes, double quotes, etc.)
2) There is to be no distinction made between upper and lower case letters. (e.g., "CAT" is the same word as "cat")
3) Blank lines are to be counted in the line numbering. (e.g., line 3 above is blank)