AC50002 Programming Languages for Data Engineering
Python Assignment

Python Assignment 1 : Ravi Mohan Lal
Student Id : 2489399

Design of Program
Google Colab Link : https://colab.research.google.com/drive/1l-SGhO6mMt_3oRIUf8wErq3UBqFf_W0h#scrollTo=4_o3hLSvWHGk

Main Features :

1. Extensively used list and dictionaries in python to manipulate data given in Values.txt and trees.txt file.
2. Also using dictionary with for loops was key to the program design and extracting key value pairs.
3. Ability to extract all abbreviations from each word.
4. Also made logic of abbreviation formation using for loop rather than the built in libraries (**itertools** — Functions creating iterators for efficient looping)
5. Google Colab notebook for easy running, without installation.
6. Extensive **Commenting** on each function for readability.
7. Use of List comprehensions, zip function, loops, conditions and dictionary indexing are the highlights.

Main Algorithm:

Step1 : defining file path in current directory for uploading.

Step2 : Saving file into list (trees.txt)

Step 3 : Preprocessing trees.txt so that hyphens and other non-characters are removed and all words Capitalized. Variable to check (***names_upper***)

Step 4: List of key value pairs from (values.txt). Varible to check (***`final list`***)

Step 5:Converting these into key value pairs by forming dictionary (dict_values)

Step 6: we change each word(tree.txt) into its characters and store in list.Variable to check (lst)

Step 7 : All abbreviation list. Variable to check (ml)

Step 8: List of all scores from each abbreviation for every work is given in (***variable check l***)

Step 9: Displaying words and their minimum scores (dict_values_final)

Some Results

```
[17]  1 #dict_values_final contains words and their minimum scores
      2 dict_values_final
```

```
{'ALDER': [15],
 'CRAB APPLE': [11],
 'COMMON ASH': [6],
 'SILVER BIRCH': [6],
 'DOWNY BIRCH': [6],
 'EUROPEAN BEECH': [6],
 'BOX': [26],
 'WILD CHERRY': [6],
 'BIRD CHERRY': [12],
 'BLACKTHORN': [6],
 'WYCH ELM': [6],
 'SMOOTHLEAVED ELM': [0],
 'COMMON HAWTHORN': [6],
 'MIDLAND HAWTHORN': [4],
 'COMMON HAZEL': [6],
 'EUROPEAN HORNBEAM': [6],
 'EUROPEAN HOLLY': [6],
 'COMMON JUNIPER': [10],
```

List of words and their minimum scores

```
1 names_abba = (zip(ml, l))
2 print(list(names_abba))
```

```
[(['ALD', 'ALE', 'ALR', 'ADE', 'ADR', 'AER'], [27, 53, 21, 47, 15, 41]), (['CRA', 'CRB',
```

List of abbreviations and their scores

```
85 print(l)
86
87
88
89
```

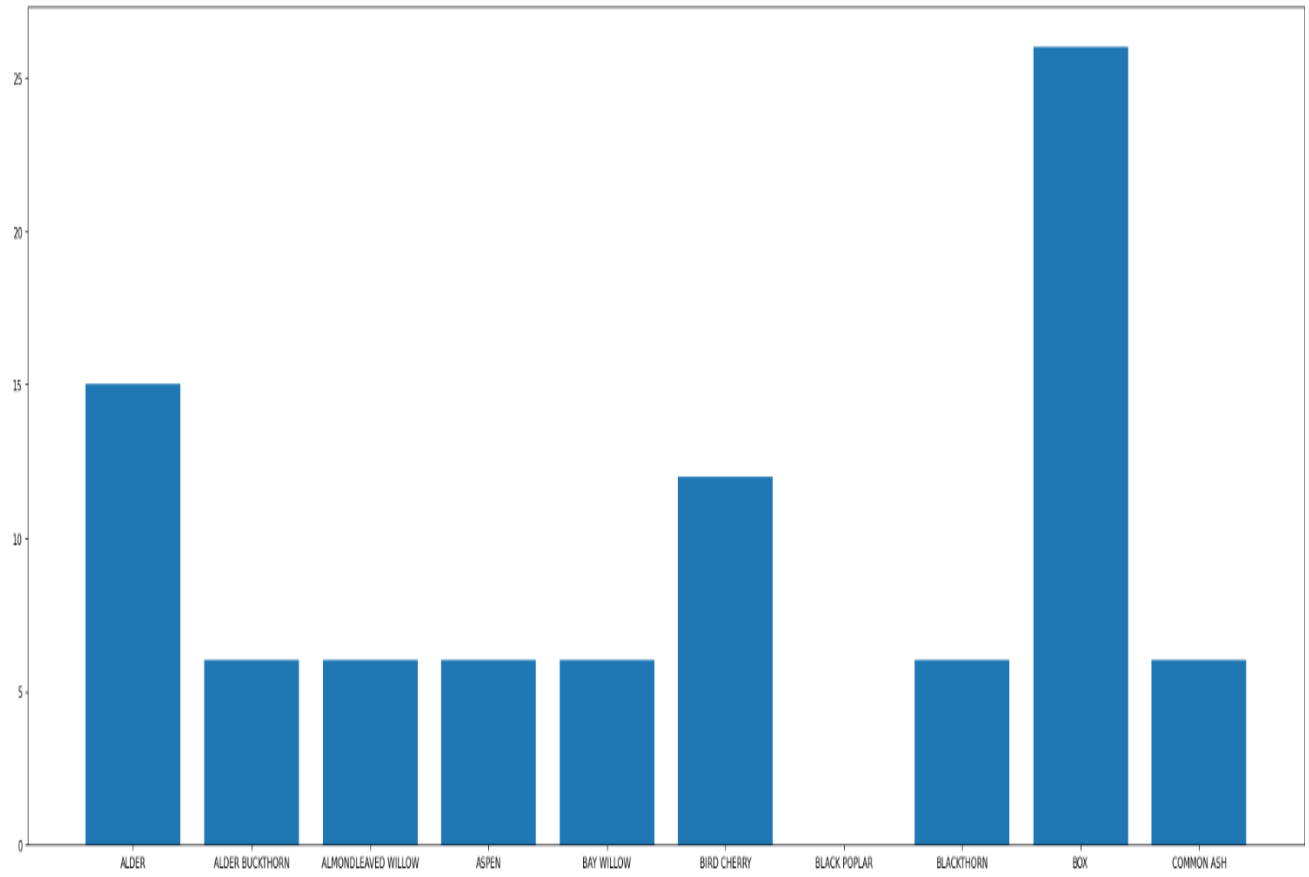
```
[[27, 53, 21, 47, 15, 41], [43, 26, 18, 43, 26, 26, 33, 36, 36, 28, 53, 36, 36, 43, 46, 11, 36, 19, 19, 26, 29, 28, 11, 11, 18, 21, 36, 36, 4
```

List of scores of abbreviations

```
19
```

```
[['ALD', 'ALE', 'ALR', 'ADE', 'ADR', 'AER'], ['CRA', 'CRB', 'CR ', 'CRA', 'CRP', 'CRP', 'CRL', 'CRE', 'CAB', 'CA ', 'CAA', 'CAP', 'CAP', 'CAL', '(
```

All abbreviations



Minimum score of each word (first 10 words) Used.