SpellChecker.py

By Jason, Srikar, and Ani

Function:

SpellChecker is a program that aids in checking a text file for spelling errors as efficiently as possible, and recommends similar words to those misspelled. It facilitates multiple languages and unicode characters, which can be chosen by the user. It allows users to upload certain file types for spell checking from their computer. After running, the program will produce a text file containing each misspelled word on a line, followed by suggestions for each word.

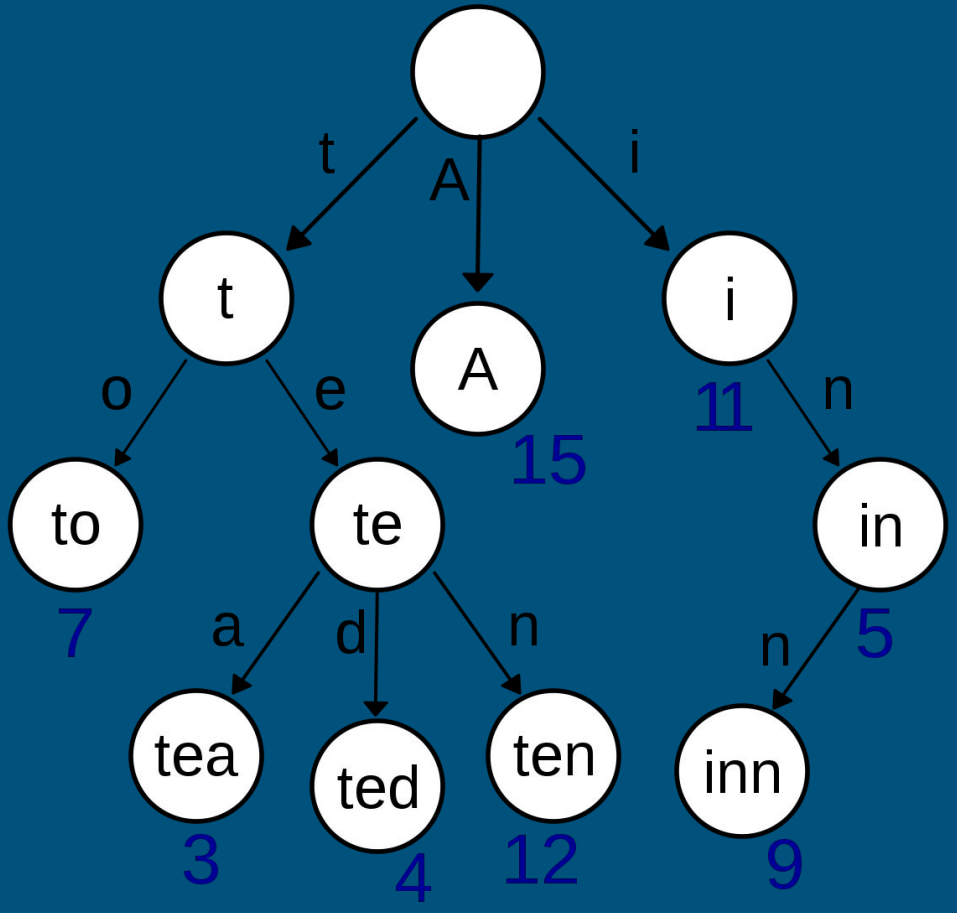
Distribution/Usage:

- Note: Program is compatible with Windows OS
- Download all files as a zip (YSP2019-Python-Final-Project-master.zip)
- Unzip the file, open the directory (SP2019-Python-Final-Project-master)
- Double-click the run.bat file
- If prompted, extract all files to a seperate file location, enter the extracted directory
- If Windows prevents the execution of run.bat, select "More Info" in the popup that appears, then "Run Anyway"
- In the SpellChecker Upload Screen, select a language by clicking a language button
- Then, click the browse button, and locate a .txt file to spell check.
- Once selected, click exit.
- After running, the program will output misspelled words and suggestions to a file named "MisspelledWords.txt"

Trie:

- SpellChecker utilizes a Trie data structure to maximise efficiency and speed
- Tries have a theoretical worst-case complexity of O(n)
  - This is the one of the best possible worst complexities attainable for any algorithm (except for O(log(n)) and O(1), which is unattainable for most tasks
  - The tradeoff is that tries can only really be used for word processing and similar tasks
- Tries, although extremely time efficient when processing long documents, can be inefficient for short documents
- Tries are generally space inefficient

Trie Diagram:



Diagnostics:

- This file is a collection of functions that would be used in the main method of the file.
- Functions:
  - The first function returns the number of words in the dictionary that we used
  - The second and third function return the dictionary and file name given the file path for the dictionary and file
  - There another 2 functions that return the time at given instant and the time between two different times
  - The last function just returns a variable called code