COMP 6651: Algorithm Design Techniques
Fall 2017: Programming Assignment 1
Due: October 8, 2017 at midnight

1 Problem

Your task is to write and test two simple spelling corrector algorithms.

You are given as input three files. The first file is called `vocab.txt`, and is a dictionary of valid words, with each word on a separate line. The number of words is at most $400,000$ and each word is at most 30 characters long. The second file is called `sentence.txt`, and contains a sentence of at most 20 words, each word being at most 30 characters long, and being separated by one or more empty spaces. The third file is called `MaxDistance.txt` and contains a single integer value k that lies between 0 and 5.

Your job is to break up the sentence into words, and for each word w in the sentence, that is *not* in the dictionary, flag it, and give a list of possible spelling corrections, that is, words in the dictionary that are at most *Levenshtein distance* k from the misspelled word. The Levenshtein distance between a word s and a word t is the number of additions, deletions, and substitutions of characters required to transform s to t.

Write your output into a file called `MisspelledWords.txt`. On each line, write a misspelled word found in the sentence, followed by a colon, and then the list of possible corrections, separated by a comma and then a space. The last correction will be followed by nothing. Each misspelled word will be on a different line. If there are no misspelled words in the sentence, the output file should simply contain the number 0.

To find the list of spelling corrections, you should implement two algorithms:

1. A simple linear search for the current word w in the dictionary, checking for each word in the dictionary if it is distance $\leq k$ from the searched word w.

2. An algorithm using BK-trees.

Some test cases will be provided on the course website. You should verify if your programs work on the test cases before submitting. Your code will also be tested on some larger input files which are not given to you in advance.

Run experiments on these two algorithms, and analyze the running time as a function of the following:

1. The size and type of dictionary.

2. The length of the search words.

3. The maximum distance value k.

2 Requirements

You must submit a zip file containing the following three items:

- Source code for the two algorithms written in C#/C++/Java on the Electronic Assignment System and
- A report with the results of your analysis.
- The data files you used in your analysis.

3 Programmer-on-duty

There will be a programmer-on-duty, Meghrig Terzian, available to help you with the assignment in the lab H-827 omn Tuesdays and Thursdays from 5-8 p.m.