Ανάκτηση Πληροφορίας - Εργαστήριο

Άσκηση 3

Όνομα: Άγγελος Τζώρτζης
Α.Μ.: 18390094

Άσκηση Α:

Κώδικας για την υλοποίηση της άσκησης:

```
# Άσκηση Α.
# Η συνάρτηση για τον υπολογισμό της απόστασης διόρθωσης δύο λέξεων.

memo = {}


def lev_dist(text1, text2):
    if text1 == "":
        return len(text2)
    if text2 == "":
        return len(text1)
    cost = 0 if text1[-1] == text2[-1] else 1

    a = (text1[:-1], text2)
    if not a in memo:
        memo[a] = lev_dist(*a)
    b = (text1, text2[:-1])
    if not b in memo:
        memo[b] = lev_dist(*b)
    c = (text1[:-1], text2[:-1])
    if not c in memo:
        memo[c] = lev_dist(*c)
    result = min([memo[a] + 1, memo[b] + 1, memo[c] + cost])

    return result


# Ενδεικτικά τρεξίματα για την συνάρτηση μας.
print("1: " + str(lev_dist("cats", "fast")))
print("2: " + str(lev_dist("Python", "Pethno")))
print("3: " + str(lev_dist("Greece", "Australia")))
print("4: " + str(lev_dist("Πανεπιστήμιο", "Σχολείο")))
print("5: " + str(lev_dist("Εργαστηριακή", "Άσκηση")))
print("6: " + str(lev_dist("Χριστούγεννα", "Πάσχα")))
```

Αποτελέσματα της εκτέλεσης του κώδικα:

```
1: 3
2: 3
3: 8
4: 11
5: 10
6: 10
```

Συμπεράσματα:
Όπως βλέπουμε όσο πιο “όμοιες” είναι οι λέξεις τόσο μικρότερη είναι η απόσταση levenshtein μεταξύ τους. Επίσης βλέπουμε πώς λειτουργεί σωστά και για την Αγγλική άλλα και για την ελληνική γλώσσα. Η μέθοδο που χρησιμοποιήσαμε είναι μία αναδρομική συνάρτηση με χρήση προσωρινής “μνήμης”. Συγκεκριμένα, υπολογίζουμε την ελάχιστη απόσταση σε κάθε υποσυμβολοσειρά των δύο που δώσαμε, μέχρι να φτάσουμε στην απόσταση των 2 συμβολοσειρών που δώσαμε. Η “μνήμη” που χρησιμοποιήσαμε είναι στην μορφή λεξικού και αποθηκεύουμε το κάθε αποτύπωμα που προκύπτει ώστε να μην χρειαστεί να υπολογιστεί ξανά, ελαχιστοποιώντας έτσι τις φορές που την καλούμε. Απο τις συναρτήσεις αναδρομής για την απόσταση διόρθωσης είναι αυτή που απαιτεί τις λιγότερες κλήσεις. Αποτελεί μια γρήγορη και αποτελεσματική μέθοδο για την εύρεση της απόστασης διόρθωσης. Αξίζει να σημειωθεί πως γίνεται να βρεθεί και με δυναμικό τρόπο χρησιμοποιώντας έναν πίνακα δύο διαστάσεων.

Άσκηση Β:
Κώδικας για την υλοποίηση της άσκησης:

```
# Άσκηση Β.
# Κατεβάζουμε αρχεία με λέξεις
import nltk
nltk.download("words")
from nltk.corpus import words

# Ορίζουμε το λεξιλόγιο μας με τις λέξεις που κατεβάσαμε από το nltk.
correct_words = words.words()

# Φτιάχνουμε μία λίστα με την λέξη που θέλουμε να διορθώσουμε.
incorrect_words = []
incorrect_word = input("Enter a word: ")
incorrect_words.append(incorrect_word)

for word in incorrect_words:
    temp = [(lev_dist(word, w), w) for w in correct_words if w[0] == word[0]]
    print(sorted(temp, key=lambda val: val[0])[0][1])
```

Αποτελέσματα της εκτέλεσης του κώδικα:

```
In [6]: runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/Us

Enter a word: happy
happy

In [7]: runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/Us

Enter a word: intelliengt
intelligent

In [8]: runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/Us

Enter a word: azmaing
aiming
```

Υ.Γ.: Αποθηκεύουμε την λέξη που εισάγουμε σε έναν πίνακα γιατί δεν αποτυπωνόταν σωστά τα αποτελέσματα μας.
Θα μπορούσαμε να χρησιμοποιήσουμε και την συνάρτηση edit_distance που μας δίνει έτοιμη ή nltk

Συμπεράσματα:
Διορθώνουμε τις λέξεις με την χρήση της lev_dist που φτιάξαμε στο προηγούμενο ερώτημα. Όπως βλέπουμε η λανθασμένη λέξη “happpy” διορθώνεται και γίνεται “happy” και το “intelliengt” γίνεται “intelligent”. Όμως το “azmaing” μετατρέπεται σε “aiming” το οποίο είναι σωστό ορθογραφικά αλλα μπορεί να θέλαμε να βγάλει την λέξη “amazing”. Οπότε η διόρθωση των λέξεων μάς μπορεί να μην έχει το επιθυμητό αποτέλεσμα και ας τις διορθώνει σωστά.

Με την εισαγωγή Ελληνικής λέξης βλέπουμε το εξής:

```
In [11]: runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/U

Enter a word: Ποδόσφαιρο
Traceback (most recent call last):

  File "C:\Users\RedTo\AppData\Local\Temp/ipykernel_20112/1234162277.py", line 1, in <module>
    runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/Users/

  File "C:\Users\RedTo\anaconda3\lib\site-packages\debugpy\_vendored\pydevd\_pydev_bundle\pydev_umd.py", l
    execfile(filename, namespace)

  File "C:\Users\RedTo\anaconda3\lib\site-packages\debugpy\_vendored\pydevd\_pydev_imps\_pydev_execfile.py
    exec(compile(contents + "\n", file, 'exec'), glob, loc)

  File "C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py", line 52, in <module>
    print(sorted(temp, key=lambda val: val[0])[0][1])

IndexError: list index out of range
```

Το σφάλμα αυτό προκύπτει καθώς δέν υπάρχουν ελληνικές λέξεις στο λεξιλόγιο που κατεβάσουμε αλλά μπορούμε να εισάγουμε όποιες λέξεις θέλουμε χειροκίνητα.

Τώρα θα δούμε το αποτέλεσμα με την εισαγωγή της λέξης ποδόσφαιρο στο λεξιλόγιο μας.

```
In [12]: runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/U
Traceback (most recent call last):

  File "C:\Users\RedTo\AppData\Local\Temp/ipykernel_20112/1234162277.py", line 1, in <module>
    runfile('C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py', wdir='C:/Users/

  File "C:\Users\RedTo\anaconda3\lib\site-packages\debugpy\_vendored\pydevd\_pydev_bundle\pydev_umd.py", l
    execfile(filename, namespace)

  File "C:\Users\RedTo\anaconda3\lib\site-packages\debugpy\_vendored\pydevd\_pydev_imps\_pydev_execfile.py
    exec(compile(contents + "\n", file, 'exec'), glob, loc)

  File "C:/Users/RedTo/Documents/Ανάκτηση Πληροφορίας/information_retrieval_lab3.py", line 43, in <module>
    correct_words = words.words()

  File "C:\Users\RedTo\anaconda3\lib\site-packages\nltk\corpus\reader\wordlist.py", line 21, in words
    for line in line_tokenize(self.raw(fileids)):

  File "C:\Users\RedTo\anaconda3\lib\site-packages\nltk\corpus\reader\api.py", line 218, in raw
    contents.append(fp.read())

  File "C:\Users\RedTo\anaconda3\lib\site-packages\nltk\data.py", line 1055, in read
    chars = self._read(size)

  File "C:\Users\RedTo\anaconda3\lib\site-packages\nltk\data.py", line 1344, in _read
    chars, bytes_decoded = self._incr_decode(bytes)

  File "C:\Users\RedTo\anaconda3\lib\site-packages\nltk\data.py", line 1375, in _incr_decode
    return self.decode(bytes, "strict")

UnicodeDecodeError: 'ascii' codec can't decode byte 0xce in position 0: ordinal not in range(128)
```

Τώρα προκύπτει σφάλμα καθώς οι Ελληνικοί χαρακτήρες δεν μπορούν να κωδικοποιηθούν. Οπότε δεν μπορούμε να πάρουμε κάποιο αποτέλεσμα.

Υπάρχουν τρόποι να μπορούμε να διορθώσουμε και Ελληνικές λέξεις εάν φτιάξουμε η βρούμε ένα λεξιλόγιο με Ελληνικές λέξεις και κωδικοποιήσουμε σωστά τους Ελληνικούς χαρακτήρες και εφαρμόσουμε την παραπάνω μέθοδο σε αυτά.