

**Σημείωση:** Ομοιότητα με φετινές εργασίες ή εργασίες παλαιότερων ετών συνεπάγεται τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων.

# Εργασία 1η - Compressed Trie

Προθεσμία Υποβολής: Τετάρτη 30 Μαρτίου, 23:59

Γράψτε ένα πρόγραμμα σε java το οποίο υλοποιεί ένα λεξικό της αγγλικής γλώσσας μέσω ενός συμπιεσμένου Trie με βάση (radix) το αγγλικό αλφάβητο (a-z). Συμπιεσμένο ονομάζεται κάθε Trie του οποίου οι κόμβοι που δεν είναι τερματικοί και έχουν μόνο ένα παιδί, ενώνονται με το παιδί τους. Οι κόμβοι που έχουν ενωθεί αντιπροσωπεύουν μία συμβολοσειρά αντί για ένα μεμονωμένο χαρακτήρα. Εποπτικό παράδειγμα για το πως διαμορφώνεται ένα συμπιεσμένο Trie [μπορείτε να βρείτε εδώ](#).

Το πρόγραμμα σας πρέπει να περιέχει μόνο τα αρχεία κλάσεων α) TrieNode.java που περιγράφει τον κόμβο, Trie.java που περιγράφει τη δομή Trie και γ) HW1.java που περιέχει τη μέθοδο main. Οι παραπάνω κλάσεις πρέπει να βρίσκονται στο πακέτο ce326.hw1.

## Περιγραφή της λειτουργίας του προγράμματος

Το πρόγραμμα σε επανάληψη εκτυπώνει τη συμβολοσειρά "**?:**" ακολουθούμενη από ένα κενό και χαρακτήρα αλλαγής γραμμής και στη συνέχεια διαβάζει από το πληκτρολόγιο και εκτελεί μία από τις ακόλουθες εντολές εκτυπώνοντας στο stdout.

| Εντολή    | Περιγραφή  | Εκτυπώσεις  |               |
|-----------|--|---|---------------|
|           |  | Επιτυχία  | Αποτυχία      |
| -i word   | Εισάγει τη λέξη word στο Trie. Επιστρέφει επιτυχία εάν η λέξη δεν υπάρχει στο Trie και εισαχθεί επιτυχώς σε αυτό. Διαφορετικά επιστρέφει αποτυχία.   | ADD word OK*  | ADD word NOK* |
| -r word   | Διαγράφει τη λέξη word από το Trie. Επιστρέφει επιτυχία εάν η λέξη υπάρχει στο Trie και διαγραφεί επιτυχώς. Διαφορετικά επιστρέφει αποτυχία.   | RMV word OK*  | RMV word NOK* |
| -f word   | Αναζητεί τη λέξη word στο Trie.  | FND word OK*  | FND word NOK* |
| -p        | Εκτυπώνει την pre-order διαπέραση του Trie. Αρχικά εκτυπώνεται η συμβολοσειρά " <b>PreOrder:</b> " και στη συνέχεια ακολουθούν οι επιμέρους συμβολοσειρές των κόμβων του δέντρου εκτυπωμένες κατά την διαπέραση preorder. Εάν ένας κόμβος είναι τερματικός εκτυπώνεται η συμβολοσειρά " <b>#</b> " αμέσως μετά το περιεχόμενο του (χωρίς κενό). Μετά από κάθε συμβολοσειρά εκτυπώνεται ένας κενός χαρακτήρας και μετά το τελευταίο κενό εκτυπώνεται ένας χαρακτήρας <del>εκτυπώνονται δύο χαρακτήρες</del> αλλαγής γραμμής.<br><b>Σημείωση:</b> Η ρίζα του Trie που δεν έχει περιεχόμενο δεν εκτυπώνεται |   |               |
| -d        | Εκτυπώνει το σύνολο των λέξεων του αποθηκευμένου λεξικού σε αλφαβητική σειρά. Αρχικά εκτυπώνεται χαρακτήρας αλλαγής γραμμής, η συμβολοσειρά " <b>***** Dictionary *****</b> " ακολουθούμενη από χαρακτήρα αλλαγής γραμμής και στη συνέχεια οι λέξεις που περιέχει το λεξικό. Μετά από κάθε λέξη εκτυπώνεται χαρακτήρας αλλαγής γραμμής, ενώ μετά την τελευταία λέξη εκτυπώνονται δύο χαρακτήρες αλλαγής γραμμής αντί ενός.   |   |               |
| -w word X | Αναζητεί στο Trie όλες τις λέξεις <u>ιδίου μήκους</u> με τη λέξη <b>word</b> οι οποίες απέχουν <u>ακριβώς X</u> χαρακτήρες από τη δοθείσα λέξη. Για παράδειγμα, <ul style="list-style-type: none"><li>• οι λέξεις boy και toy απέχουν ένα χαρακτήρα.</li><li>• οι λέξεις small και smell απέχουν ένα χαρακτήρα.</li><li>• οι λέξεις small και smile απέχουν δύο χαρακτήρες.</li><li>• οι λέξεις newspaper και newsgroup απέχουν πέντε χαρακτήρες.</li></ul>  | Εκτυπώνεται χαρακτήρας αλλαγής γραμμής και η συμβολοσειρά " <b>Distant words of W (X):</b> " ακολουθούμενη από χαρακτήρα αλλαγής γραμμής, που W η λέξη word και X η απόσταση των αναζητούμενων λέξεων.<br><br>Στη συνέχεια εκτυπώνονται με αλφαβητική σειρά οι λέξεις που απέχουν X χαρακτήρες από την δοθείσα. Μετά από κάθε λέξη εκτυπώνεται χαρακτήρας αλλαγής γραμμής, ενώ μετά την τελευταία λέξη εκτυπώνονται δύο χαρακτήρες αλλαγής γραμμής αντί ενός. |               |

**Σημείωση:** Ομοιότητα με φειτινές εργασίες ή εργασίες παλαιότερων ετών συνεπάγεται τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων.

|           |   |  |
|-----------|---|--|
|           |   |  |
| -s suffix | Αναζητεί στο Trie λέξεις που περιέχουν τη δοθείσα κατάληξη suffix.  | Εκτυπώνεται χαρακτήρας αλλαγής γραμμής και η συμβολοσειρά "Words with suffix S:" ακολουθούμενη από χαρακτήρα αλλαγής γραμμής, όπου S η δοθείσα κατάληξη.<br>Στη συνέχεια εκτυπώνονται με αλφαβητική σειρά οι λέξεις που περιέχουν τη δοθείσα κατάληξη. Μετά από κάθε λέξη εκτυπώνεται χαρακτήρας αλλαγής γραμμής, ενώ μετά την τελευταία λέξη εκτυπώνονται δύο χαρακτήρες αλλαγής γραμμής αντί ενός. |
| -q        | Εκτυπώνει τη συμβολοσειρά "Bye bye!" ακολουθούμενη από χαρακτήρα αλλαγής γραμμής και τερματίζει το πρόγραμμα. |  |

\* Μετά από κάθε μήνυμα εκτυπώνεται χαρακτήρας αλλαγής γραμμής.

## Περιορισμοί - Παρατηρήσεις

- Όλες οι λέξεις που εισάγονται ή αναζητούνται πρέπει να περιέχουν μόνο γράμματα από το αγγλικό αλφάβητο (όχι σημεία στίξης, αριθμητικά ψηφία ή γράμματα άλλου αλφαβήτου). Εάν μία λέξη δεν υπακούει στον παραπάνω περιορισμό αγνοείται και επιστρέφεται μήνυμα αποτυχίας ο κώδικας συνεχίζει στην επόμενη επανάληψη.
- Οι λέξεις ή τα περιεχόμενα των κόμβων εκτυπώνονται πάντοτε με πεζά γράμματα. Ο τρόπος αποθήκευσης τους στη δομή (πεζά ή κεφαλαία) αφήνεται σε εσάς.
- Οι λέξεις προς εισαγωγή μπορεί να περιέχουν και κεφαλαία και πεζά γράμματα (π.χ. sMaLI, SMALL ή small).

## Διάβασμα από το πληκτρολόγιο

Ο παρακάτω κώδικας χρησιμοποιεί την κλάση [java.util.Scanner](#) για να διαβάσει σε επανάληψη από το πληκτρολόγιο. Μην αντιγράψετε τον παρακάτω κώδικα μέσω copy-paste, διότι εισάγονται χαρακτήρες που δεν είναι ASCII. Για το πρόγραμμα σας είναι βολικότερο το διάβασμα λέξη-λέξη.

| Διάβασμα λέξη-λέξη  | Διάβασμα γραμμή-γραμμή  |
|---|---|
| <pre>import java.util.Scanner;  public class ReadWord {      public static void main(String []args) {         Scanner sc = new Scanner(System.in);         String word;         while(sc.hasNext()) {             word = sc.next();         }     } }</pre> | <pre>import java.util.Scanner;  public class ReadLine {      public static void main(String []args) {         Scanner sc = new Scanner(System.in);         String line;         while(sc.hasNextLine()) {             line = sc.nextLine();         }     } }</pre> |

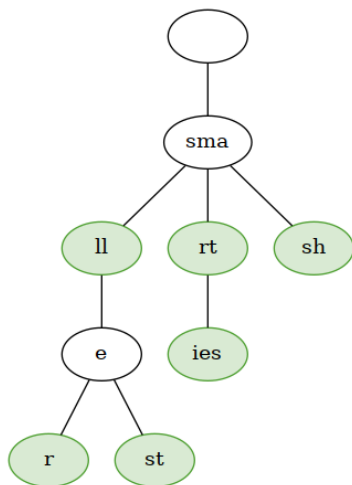
## Εκτύπωση του Trie σε εικόνα (μόνο για debugging)

Η εκτύπωση του Trie σε εικόνα δεν ζητείται από την εργασία. Μπορείτε όμως να την υλοποιήσετε για να διευκολύνετε την αποσφαλμάτωση του κώδικα σας.

Για την εκτύπωση ενός γραφήματος σε εικόνα μπορείτε να χρησιμοποιήσετε το πρόγραμμα [dot](#) της σουίτας [graphviz](#). Για να το κάνετε αυτό αρκεί να δημιουργήσετε ένα αρχείο κειμένου (με κατάληξη .dot στο όνομα του) που να περιγράφει τη δομή με βάση τις απαιτήσεις του παραπάνω προγράμματος. Η σειρά εκτύπωσης των

**Σημείωση:** Ομοιότητα με φεινές εργασίες ή εργασίες παλαιότερων ετών συνεπάγεται τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων.

κόμβων του δέντρου είναι η pre-order διαπέραση του. Η εμφάνιση μιας ακμής μπορεί να προηγείται της εμφάνισης του κόμβου του δέντρου μέσα στο αρχείο.



Για παράδειγμα, για το διπλανό Trie με εγγραφές small, smaller, smallest, smart, smarties και smash μπορείτε να κατεβάσετε [το ενδεικτικό αρχείο dot εδώ](#).

Στο παραπάνω αρχείο, κάθε κόμβος έχει ένα μοναδικό αναγνωριστικό που αποτελείται από έναν ακέραιο. Στο αρχείο, ο ακέραιος που αφορά τον κόμβο με περιεχόμενα "sma" είναι 1418481495.

Ο μοναδικός ακέραιος για κάθε κόμβο προκύπτει από την επιστρεφόμενη τιμή της συνάρτησης hashCode(). Η συνάρτηση hashCode() υπάρχει σε όλες τις κλάσεις.

Αφού γράψετε το αρχείο dot μπορείτε να δημιουργήσετε μία εικόνα PNG με την προϋπόθεση ότι έχετε εγκαταστήσει τη σουίτα graphviz στον υπολογιστή σας. Η εντολή μετασχηματισμού είναι η εξής:

```
$> dot -Tpng input_filename.dot -o output_filename.png
```

Όπου είναι `input_filename.dot` το όνομα του αρχείου dot που φτιάξατε προηγούμενα και `output_filename.png` είναι το όνομα της τελικής εικόνας που θα προκύψει.

## Εκτύπωση συμβολοσειράς σε αρχείο

Το παρακάτω πρόγραμμα εκτυπώνει τη συμβολοσειρά fileContents στο αρχείο με όνομα fileName το οποίο αν δεν υπάρχει το δημιουργεί και εάν υπάρχει διαγράφει τα περιεχόμενα του.

```
import java.io.PrintWriter;
import java.io.FileNotFoundException;

public class PrintTxt2File {
    void printDot(String fileContents, String fileName) {
        try(PrintWriter wr = new PrintWriter(new File(fileName))) {
            wr.print(fileContents);
        }
        catch(FileNotFoundException ex) {
            System.out.print("Unable to write file '"+fileName+"'");
        }
    }
}
```

## Οδηγίες Αποστολής

Η αποστολή της εργασίας θα γίνει μέσω της πλατφόρμας autolab. Για την υποβολή ακολουθήστε τα εξής βήματα:

- Συμπίεστε σε μορφή zip, από το project σας τα περιεχόμενα του καταλόγου `src/ce326/hw1/`, που είναι τα τρία αρχεία java της συγκεκριμένης εργασίας. Το αρχείο που προκύπτει πρέπει να έχει όνομα `hw1.zip`.
- Συνδέστε στο [autolab](#) (μέσω VPN) και επιλέγετε το μάθημα **ECE326\_2022 (S22)** και από αυτό την εργασία HW1.

**Σημείωση:** Ομοιότητα με φετινές εργασίες ή εργασίες παλαιότερων ετών συνεπάγεται τον αυτόματο μηδενισμό στο μάθημα όλων των εμπλεκόμενων.

- Για να υποβάλετε την εργασία σας κάνετε click στην επιλογή “I affirm that I have compiled with this course academic integrity policy...” και πατάτε submit. Στη συνέχεια επιλέγετε το αρχείο **hw1.zip** που δημιουργήσατε παραπάνω.