Autocomplete.Me

Γράψτε ένα πρόγραμμα σε JAVA το οποίο υλοποιεί την λειτουργία autocomplete ως εξής:

1. Διαβάζει λέξεις από αρχεία που σας δίνονται δημιουργώντας μία βάση δεδομένων με προτεινόμενες λέξεις
2. Ενθέτει κάθε λέξη που διαβάζει σε μία δομή λεξικού η οποία αποτελεί και τη βάση πρότασης λέξεων από το πρόγραμμα.
3. Αφού ολοκληρώσει την διαδικασία διαβάσματος το πρόγραμμα έχει τη δυνατότητα να προτείνει στο χρήστη λέξεις με βάση το πρόθεμα λέξης (αλφαριθμητικό) που αυτός εισάγει κάθε φορά.
4. Αποθήκευση και ανάκτηση του λεξικού σε δυαδική μορφή.

Διαδικασία διαβάσματος από αρχείο

Σας δίνονται αρχεία κειμένου από το Project Gutemberg τα οποία προέρχονται από βιβλία. Διαβάζετε τα αρχεία κειμένου ως εξής:

1. Διαβάζετε λέξη-λέξη.
2. Εάν μία λέξη ξεκινά με κεφαλαίο γράμμα ή περιέχει μόνο κεφαλαία την καταχωρείτε ως έχει (για τη μοναδική εξαίρεση δείτε παρακάτω).
3. Εάν μία λέξη περιέχει σημείο στίξης στο τέλος της το αφαιρείτε και την καταχωρείτε. Επίσης εάν η επόμενη λέξη από το σημείο στίξης ξεκινά με κεφαλαίο, δεν θα καταχωρηθεί με το πρώτο γράμμα της κεφαλαίο αλλά με μικρό.
4. Δεν καταχωρείτε λέξεις (τις αγνοείτε) που περιέχουν σημεία στίξης στην μέση (π.χ. don’t, wasn’t κλπ).

Το πρόγραμμα σας θα πρέπει να έχει τη δυνατότητα να διαβάζει ένα ή περισσότερα αρχεία. Για κάθε αρχείο που διαβάζει ενθέτει την πληροφορία στην υφιστάμενη δομή λεξικού (δες παρακάτω).

Αρχεία κειμένου (ανανεώθηκε 15/3)

Ένθεση των λέξεων σε δομή λεξικού

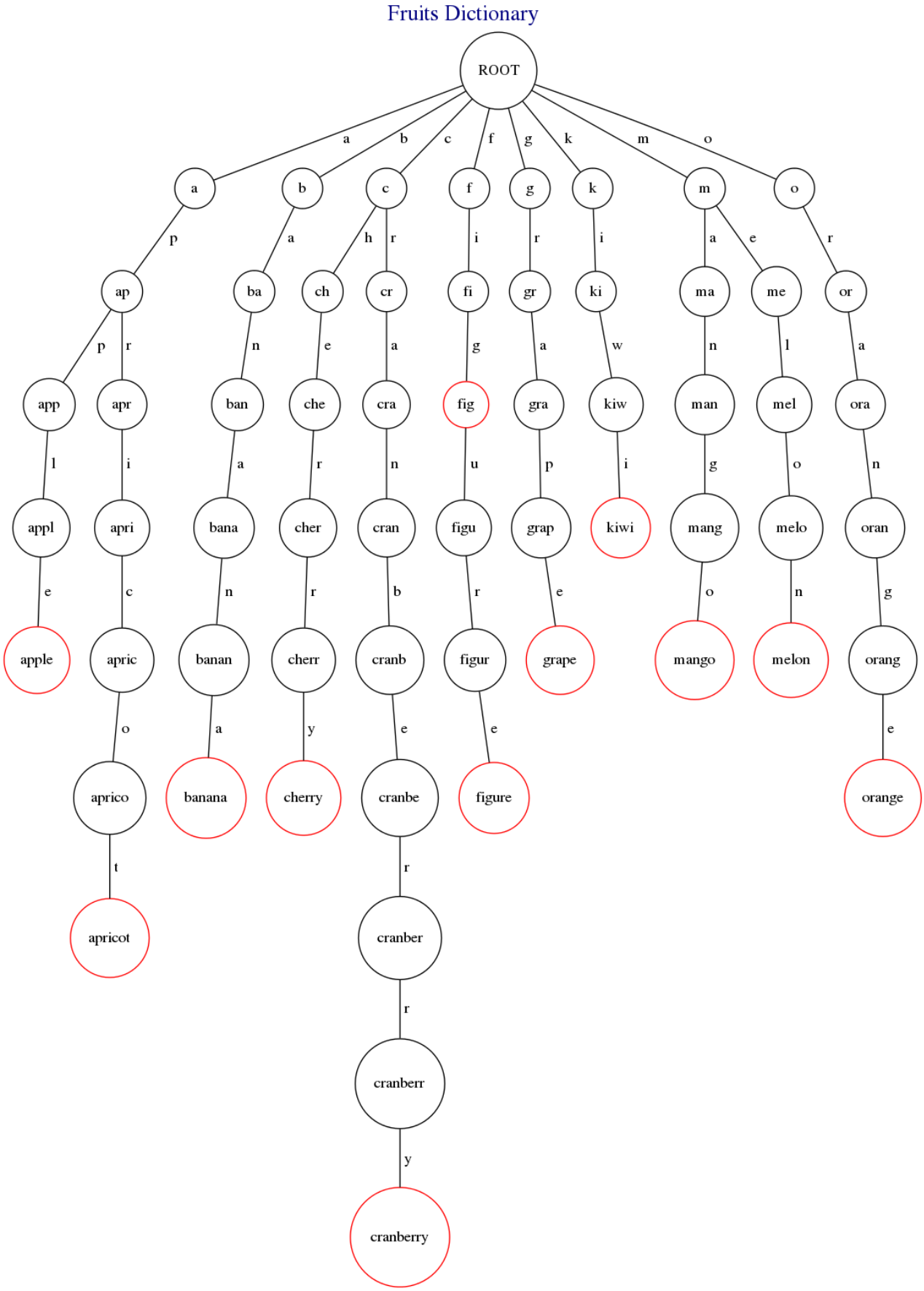
Μία δενδρική δομή λεξικού είναι μία δομή δεδομένων κάθε κόμβος της οποίας περιέχει τα εξής:

1. ένα πίνακα **N** δεικτών προς τα παιδιά του, όπου **N** είναι ο αριθμός των γραμμάτων του λεξικού (το αγγλικό αλφάβητο που μας ενδιαφέρει έχει **26** γράμματα).
2. μία μεταβλητή τύπου **boolean** που δηλώνει εάν ο κόμβος αντιστοιχεί στο τελευταίο γράμμα μιας λέξης ή όχι (αποτελεί τερματικό κόμβο λέξης).
3. προαιρετικά (για το σκοπό της παρούσας εργασίας), μία μεταβλητή τύπου **enum** που ο εφόσον ο κόμβος είναι τερματικός δηλώνει εάν η λέξη περιέχει μόνο μικρά γράμματα, μόνο κεφαλαία γράμματα ή το πρώτο κεφαλαίο και όλα τα υπόλοιπα μικρά.

Η ρίζα του δένδρου δεν αντιστοιχεί σε κάποιο γράμμα του λεξικού. Δείχνει όμως στο πρώτο γράμμα κάθε λέξης.

Παρακάτω δίνεται η γραφική αναπαράσταση μίας δομής λεξικού για το παρακάτω σύνολο λέξεων:
apple, apricot, banana, orange, mango, melon, fig, figure, grape, cherry, cranberry, kiwi.
Με κόκκινο χρώμα σημειώνονται οι τερματικοί κόμβοι της δομής. Παρατηρήστε ότι για τις λέξεις **apple** και **apricot** το μονοπάτι **ap** που αντιστοιχεί στα δύο πρώτα γράμματα είναι κοινό. Ανάλογα για τις

λέξεις **fig** και **figure**. Σε κάθε κόμβο (τερματικό ή μη τερματικό) εμφανίζεται η λέξη ή το τμήμα της λέξης στην οποία αντιστοιχεί. Η πληροφορία αυτή εξάγεται από τη δομή λεξικού, ΔΕΝ αποθηκεύεται όμως σε αυτή με τον τρόπο που εμφανίζεται στη γραφική απεικόνιση.



Επίλυση της εργασίας με χρήση άλλων δομών εκτός της προτεινόμενης (π.χ. συνδεδεμένη λίστα) γίνονται δεκτές με ισχυρό penalty (>50%)

Προτάσεις λέξεων με βάση συγκεκριμένο πρόθεμα

Αφού ολοκληρώσετε το “χτίσιμο” της δομή λεξικού, το πρόγραμμα σας είναι έτοιμο να προτείνει λέξεις προς τον τελικό χρήστη (λειτουργία suggest). Οι προτάσεις έχουν ως εξής.

Εάν στο παραπάνω παράδειγμα με τα φρούτα ο χρήστης:

- εισάγει **ap** τότε θα εμφανιστούν οι λέξεις **apple** και **apricot**.
- Εάν εισάγει **app** θα εμφανιστεί **apple**.
- Εάν εισάγει **m** θα εμφανιστούν **mango** και **mellon**.

Εάν υπάρχουν λέξεις που ξεκινούν από το εισαγόμενο πρόθεμα, αλλά έχουν το πρώτο γράμμα κεφαλαίοι ή όλα τα γράμματα κεφαλαία εμφανίζονται όπως έχουν αποθηκευτεί στη δομή λεξικού.

Αποθήκευση και ανάκτηση του λεξικού

Το πρόγραμμα σας θα πρέπει να έχει την δυνατότητα να αποθηκεύει το λεξικό σε αρχείο σε δυαδική μορφή μέσω ενός ή περισσότερων Serialized αντικειμένων. Αντίστοιχα, θα μπορεί να φορτώνει οποιοδήποτε λεξικό με την παραπάνω μορφή αντικαθιστώντας το υφιστάμενο λεξικό με ένα νέο. Μέσω της παραπάνω λειτουργικότητας θα είναι εφικτό να τερματίσετε το πρόγραμμα αφού αποθηκεύσετε το λεξικό και στη συνέχεια αφού επανεκκινήσετε το πρόγραμμα να φορτώσετε ξανά το λεξικό από δυαδικό αρχείο.

Περιγραφή του μενού προς τον χρήστη

Το πρόγραμμα που θα γράψετε θα πρέπει να έχει το εξής μενού προς τον τελικό χρήστη:

```
----------- MENU -----------
   1. Load dictionary from binary file (type: load fromFilepath)
   2. Save dictionary to binary file (type: save toFilepath)
   3. Populate dictionary from txt file (type: read fromTxtFilePath)
   4. Suggest word (type: suggest wordPhrase)
   5. Print dictionary information (type: print)
   6. Quit (type: quit)
```

Η επιλογή 5 (Print dictionary) εκτυπώνει σε ένα αρχείο κειμένου το σύνολο των λέξεων που περιέχονται στο λεξικό, μία λέξη σε κάθε σειρά. Στο τέλος εμφανίζει το όνομα του αρχείου κειμένου που δημιουργήθηκε και τον αριθμό των λέξεων που καταγράφηκαν σε αυτό.

Αφαίρεση σημείων στίξης

Η αφαίρεση των σημείων στίξης μπορεί να γίνει με τον παρακάτω κώδικα

```
import java.util.Scanner;
import java.io.File;

import java.util.regex.*;

public class ReadWithScanner {
  public static void main(String []args) {

    try {
      Scanner sc = new Scanner(new File("011.txt"));

      while( sc.hasNext() ) {
        String word = sc.next();


        Pattern p = Pattern.compile("\\p{Punct}");
        Matcher m = p.matcher(word);
        if(m.find()) {
          word = word.substring(0,m.start());
        }
        word = word.trim();
        System.out.println(word);
      }
    } catch(Exception ex) {
      ex.printStackTrace();
    }
  }
}
```

Τρόπος Αποστολής

Πακετάρετε **ΜΟΝΟ** τα αρχεία java αφού τα συμπιέσετε σε ένα αρχείο ZIP με όνομα το όνομα και το ΑΕΜ κάθε μέλους της ομάδας σας (*π.χ. GiorgosThanos_1234_PeterGordon_1235.zip*). Τα αρχεία σας θα πρέπει να περιέχονται στην ιεραρχία καταλόγων που ορίζεται από το **package ce325.hw1**. Αποστείλετε την δουλειά σας με e-mail στην διεύθυνση **ce325.course@gmail.com** ως εξής:

- Τίτλος (subject): **CE325 hw01**
- συνημμένο το παραπάνω αρχείο ZIP.
- Στο σώμα του μηνύματος τα ονόματα και τα ΑΕΜ της ομάδας σας.

Εργασίες που δεν είναι συνεπείς με τους παραπάνω περιορισμούς δεν αξιολογούνται.