Πανεπιστήμιο Ιωαννίνων – Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Δομές Δεδομένων [ΜΥΥ303] – Χειμερινό Εξάμηνο 2023

7η Εργαστηριακή Άσκηση
Ψηφιακά Δένδρα
Παράδοση έως Πέμπτη 21/12,
23:55 από το eCourse

ΠΡΟΣΟΧΗ: Γράψτε σε κάθε αρχείο που παραδίδετε τα ονόματα, τους Α.Μ. των μελών της ομάδας σας, καθώς και το group του εργαστηρίου σας. Συμπεριλάβετε όλα τα αρχεία σας (κώδικας Java και lab7results.txt) σε ένα zip αρχείο. Το όνομα που θα δώσετε στο συμπιεσμένο αρχείο θα αποτελείται από το group του εργαστηρίου στο οποίο έχετε τοποθετηθεί καθώς και από το ID της ομάδας σας (π.χ., G1_ID1.zip).

Το πρόγραμμα StringTrie.java υλοποιεί ένα ψηφιακό δένδρο (trie) το οποίο αποθηκεύει λέξεις, δηλαδή αντικείμενα τύπου String όπου ο κάθε χαρακτήρας είναι γράμμα του λατινικού αλφάβητου 'a'-'z'. Κάθε κόμβος x του trie περιέχει ένα πίνακα Node next[R] συνδέσμων σε R = 26 παιδιά, όπου ο σύνδεσμος x.next[j] αντιστοιχεί στο γράμμα ('a' + j). Δηλαδή ο x.next[0] στο 'a', ο x.next[1] στο 'b', ..., και ο x.next[25] στο 'z'. Κάθε κόμβος x του trie περιλαμβάνει επίσης μια μεταβλητή σήμανσης boolean mark, όπου x.mark == true αν ο x αντιστοιχεί στο τέλος μιας λέξης που έχει εισαχθεί στη δομή.

Το StringTrie.java περιλαμβάνει τις ακόλουθες μεθόδους:

boolean *contains*(String s)	Επιστρέφει true αν η λέξη s είναι αποθηκευμένη στο trie, και false διαφορετικά.
void *insert*(String s)	Εισάγει τη λέξη s στο trie.
void *delete*(String s)	Διαγράφει τη λέξη s από το trie.
Iterable ⟨String⟩ *words*()	Επιστρέφει μια συλλογή με όλες τις λέξεις του trie σε λεξικογραφική σειρά.
void *printWords*(Iterable ⟨String⟩ S)	Τυπώνει όλες τις λέξεις στη συλλογή S.

Ζητούμενες Μέθοδοι

Τροποποιήστε κατάλληλα το πρόγραμμα StringTrie.java έτσι ώστε να αποθηκεύει το πλήθος των εμφανίσεων της κάθε λέξης. Για το σκοπό αυτό, μπορείτε να χρησιμοποιήσετε τη μεταβλητή int count που έχουμε σε κάθε κόμβο του δένδρου και να κάνετε τις κατάλληλες αλλαγές στις μεθόδους *contains*, *insert*, *delete* και *printWords*:

int *contains*(String s)	Επιστρέφει το πλήθος των εμφανίσεων της λέξης s. Αν η s δεν έχει εισαχθεί στο trie τότε επιστρέφει μηδέν.
void *insert*(String s)	Εισάγει τη λέξη s στο trie. Αν η s υπάρχει ήδη στο trie τότε αυξάνει τον αντίστοιχο μετρητή κατά ένα.

void $delete$(String s)	Μειώνει τον μετρητή εμφανίσεων της λέξης s κατά ένα. Αν ο μετρητής μηδενιστεί τότε διαγράφει τη λέξη s από το trie.
void $printWords$(Iterable⟨Item⟩ S)	Τυπώνει όλες τις λέξεις της συλλογής S, μαζί με το πλήθος των εμφανίσεων τους.

Στη μέθοδο $printWords$, καθώς και στις μεθόδους $wordsWithPrefix$ και $wordsThatMatch$ που ορίζουμε στη συνέχεια, χρησιμοποιούμε τη βοηθητική κλάση

```
private static class Item {
    private String s;
    private int count;
}
```

όπου αποθηκεύουμε μια λέξη μαζί με το πλήθος των εμφανίσεων της.

Επιπλέον, συμπληρώστε στο πρόγραμμα StringTrie.java τις παρακάτω μεθόδους:

String $longestPrefixOf$(String s)	Επιστρέφει τη μεγαλύτερη λέξη που είναι αποθηκευμένη στο trie, η οποία είναι πρόθεμα της λέξης s.
Iterable ⟨Item⟩ $wordsWithPrefix$(String s)	Επιστρέφει μια συλλογή με όλες τις λέξεις (και το πλήθος των εμφανίσεων τους), σε λεξικογραφική σειρά, οι οποίες είναι αποθηκευμένες στο trie και έχουν ως πρόθεμα τη λέξη s.
int $countWordsWithPrefix$(String s)	Επιστρέφει το πλήθος των διαφορετικών λέξεων (και όχι των εμφανίσεων τους) που είναι αποθηκευμένες στο trie και έχουν ως πρόθεμα τη λέξη s.
Iterable ⟨Item⟩ $wordsThatMatch$(String s)	Επιστρέφει μια συλλογή με όλες τις λέξεις (και το πλήθος των εμφανίσεων τους), σε λεξικογραφική σειρά, οι οποίες είναι αποθηκευμένες στο trie και ταιριάζουν με τη λέξη s. Η s μπορεί να περιέχει τον ειδικό χαρακτήρα '?' ο οποίος ταιριάζει οποιοδήποτε γράμμα 'a'-'z'.
Item $mostFrequent$()	Επιστρέφει ένα Item I, που περιέχει την λέξη $I.s$ του trie με το μέγιστο πλήθος εμφανίσεων $I.count$.

Υπόδειξη: Για την υλοποίηση της longestPrefixOf(s) βασιστείτε στον κώδικα της μεθόδου contains, ενώ για τις wordsWithPrefix(s) και wordsThatMatch(s) βασιστείτε στον κώδικα της μεθόδου words. Τέλος, στη countWordsWithPrefix(s) θα πρέπει ο χρόνος εκτέλεσης να είναι ανάλογος του πλήθους των χαρακτήρων στο String s.

Εκτέλεση Προγραμμάτων

Η main μέθοδος διαβάζει από ένα αρχείο εισόδου μία ακολουθία λέξεων, τις οποίες εισάγει σε ένα trie. Για την εκτέλεση του προγράμματος θα χρησιμοποιήσουμε για αρχείο εισόδου το DickensB.txt, το οποίο περιέχει πάνω από 5000000 λέξεις της αγγλικής, από τις οποίες περίπου οι 70000 είναι διαφορετικές. (Το αρχείο αυτό περιέχει μόνο λέξεις με μικρά γράμματα.)

Για την εκτέλεση του προγράμματος γράψτε

java StringTrie < DickensB. txt

Μετά την κατασκευή του ψηφιακού δένδρου, η main εκτελεί τις κλήσεις

- contains(“astonished”)
- contains(“carol”)
- contains(“pigeon”)
- contains(“wondered”)
- longestPrefixOf(“governmental”)
- countWordsWithPrefix(”caro”) και wordsWithPrefix("caro")
- countWordsWithPrefix(”sco”)
- wordsThatMatch("sc????e") και
- mostFrequent().

Στη συνέχεια εκτελεί

- delete(“carol”)
- delete(“carouse”) και ξανά
- countWordsWithPrefix(”caro”) και wordsWithPrefix("caro").

Αποθηκεύστε τα αποτελέσματα της εκτέλεσης του προγράμματος στο αρχείο lab7results. txt.

Παραδοτέα

Ανεβάστε στο eCourse ένα zip αρχείο με το τελικό πρόγραμμα σας StringTrie.java, τα έτοιμα προγράμματα In.java και Queue.java, καθώς και με το αρχείο των αποτελεσμάτων lab7results. txt.

Το zip αρχείο πρέπει να έχει όνομα που περιλαμβάνει το group του εργαστηρίου στο οποίο έχετε τοποθετηθεί καθώς και από το ID της ομάδας σας (π.χ., G1_ID1.zip).