Πανεπιστήμιο Πειραιώς – Τμήμα Ψηφιακών Συστημάτων
Δομές Δεδομένων 2023-2024 – 2η Εργασία

Χρήστος Δουλκερίδης Ορέστης Τελέλης

Οι διανυσματικές αναπαραστάσεις λέξεων (word embeddings[1]) είναι ιδιαίτερα δημοφιλείς για την εύρεση λέξεων που σχετίζονται εννοιολογικά. Μέσω των word embeddings, λέξεις όπως *king* και *queen* έχουν μικρή απόσταση, παρόλο που λεξικογραφικά διαφέρουν πολύ. Κάθε λέξη αναπαρίσταται με ένα διάνυσμα σε έναν χώρο υψηλής διάστασης, και λέξεις που βρίσκονται κοντά στο διανυσματικό χώρο αναμένεται να έχουν όμοιο νόημα. Στην εργασία αυτή, θα κατασκευαστεί δομή δεδομένων που επιτρέπει την γρήγορη ανάκτηση των πιο όμοιων λέξεων με κάποια δοθείσα λέξη.

Τεχνική Περιγραφή. Καλείστε να υλοποιήσετε μεθόδους της κλάσης **`HashedHeaps`**, που αναπαριστά ***πίνακα κατακερματισμού*** που περιέχει ***σωρούς***. Οι μέθοδοι περιγράφονται παρακάτω. Ακολουθεί η δεδομένη προδιαγραφή των κλάσεων **`HashedHeaps`** και **`Node`**, μαζί με ήδη υλοποιημένες μεθόδους, που περιλαμβάνονται στο δεδομένο αρχείο **`HashedHeaps.java`**

```
class HashedHeaps {
    private Node[][] words;
    private int m;
    private int itemsInHash;
    private int k;
    private int[] itemsInHeap;

    public HashedHeaps(int sz1, int sz2)
    public void load(String sFile)
    private int hashFunc(String w)
}
```

```
class Node {
        public String word;
        public double dist;

        public Node(String word, double
            dist)
}
```

Ο $m \times k$ πίνακας **`words`** υλοποιεί τον πίνακα κατακερματισμού. Το μέγεθος του πίνακα κατακερματισμού είναι **`m`**, ενώ κάθε σωρός (heap) μπορεί να περιέχει μέχρι **`k`** στοιχεία. Η μεταβλητή **`itemsInHash`** ($\leqslant m$) καταγράφει το πλήθος των στοιχείων που περιέχονται στον πίνακα κατακερματισμού. Η μέθοδος **`hashFunc(String w)`** επιστρέφει τη θέση του πίνακα **`words`** στην οποία κατακερματίζεται η λέξη **`w`**. Ως μέθοδος επίλυσης συγκρούσεων χρησιμοποιείται η γραμμική διερεύνηση.

Η μεταβλητή **`words[i]`** αναφέρεται σε έναν απλό πίνακα που υλοποιεί έναν σωρό στη θέση i του πίνακα. Η μεταβλητή **`itemsInHeap[i]`** ($\leqslant k$) είναι το πλήθος των στοιχείων που περιέχει ο σωρός που βρίσκεται στη θέση i. Ο σωρός που υλοποιείται είναι σωρός ελαχίστων. Η ρίζα είναι **`Node(w, -1)`**, όπου **`w`** η λέξη για την οποία αποθηκεύουμε άλλες όμοιες λέξεις. Κάθε άλλη καταχώρηση **`Node(wi, dist)`** του σωρού σηματοδοτεί ότι η λέξη **`wi`** έχει απόσταση **`dist`** από τη λέξη **`w`**.

Δίνεται επιπλέον αρχείο εισόδου (data.txt) που περιέχει τις αποστάσεις 1000 λέξεων. Κάθε γραμμή του αρχείου είναι της μορφής: **`word1,word2,dist`**. Η μέθοδος **`load(String sFile)`** που δίνεται υλοποιημένη, διαβάζει τα περιεχόμενα του αρχείου και εκτελεί διαδοχικές κλήσεις της μεθόδου **`insert (String w1, String w2, double dist)`** ώστε να φορτώσει τα δεδομένα στη δομή. Για την αρχικοποίηση της δομής ώστε να μπορεί να αποθηκεύσει όλα τα δεδομένα του δοθέντος αρχείου, θα πρέπει να καλέσετε: **`HashedHeaps h = new HashedHeaps(1000, 1000);`**.

Μέθοδοι προς Υλοποίηση. Θα υλοποιήσετε τις παρακάτω μεθόδους της κλάσης **`HashedHeaps`**.

`public void insert(String w1, String w2, double dist)`: αποθηκεύει μια καταχώρηση για τη λέξη **`w1`**. Αρχικά βρίσκει τη θέση κατακερματισμού της **`w1`** στον πίνακα **`words`**. Εάν η λέξη **`w1`** υπάρχει ήδη στον πίνακα, τότε απλά προστίθεται η καταχώρηση **`Node(w2, dist)`** στο σωρό που αντιστοιχεί

[1] https://en.wikipedia.org/wiki/Word_embedding

στη λέξη `w1`. Διαφορετικά εισάγονται στο σωρό: (α) μια καταχώρηση `Node(w1, -1)`, που σηματοδοτεί ότι ο σωρός αφορά τη λέξη `w1` και αναγκάζει την καταχώρηση αυτή να είναι στη ρίζα του σωρού (αφού πρόκειται για σωρό ελαχίστων) και (β) η καταχώρηση `Node(w2, dist)`.

`public String findMostSimilarWord(String w)`: αναζητά τη λέξη που έχει τη μικρότερη απόσταση από τη λέξη `w` και την επιστρέφει. Δεν προκαλεί καμία μεταβολή στα περιεχόμενα της δομής.

`public String removeMostSimilarWord(String w)`: εξάγει τη λέξη που έχει τη μικρότερη απόσταση από τη λέξη `w` και την επιστρέφει.

`public boolean haveCommonSimilarWord(String w1, String w2, int n)`: επιστρέφει `true` είτε όταν (α) η λέξη `w1` περιέχει τη λέξη `w2` στις `n` λέξεις με τη μικρότερη απόσταση, ή (β) η λέξη `w2` περιέχει τη λέξη `w1` στις `n` πιο κοντινές λέξεις. Διαφορετικά επιστρέφει `false`. Δεν προκαλεί καμία μεταβολή στα περιεχόμενα της δομής.

Επισημάνσεις:

- ***Δεν θα χρησιμοποιήσετε `ArrayList`, ή άλλες δομές βιβλιοθήκης της Java.***
- ***Δεν θα τροποποιήσετε τους δεδομένους ορισμούς κατά οποιονδήποτε τρόπο.***

Δεν ζητείται υλοποίηση `main`. Όμως, συστήνεται να τρέξετε δοκιμές στις υλοποιήσεις σας, ώστε να βεβαιωθείτε ότι δουλεύουν ορθά.

Διαδικαστικά Θέματα

ΠΑΡΑΔΟΤΕΟ. Θα παραδώσετε αρχείο ΑΜ_Επώνυμο_Όνομα.zip (όπου ΑΜ ο αριθμός μητρώου) που θα περιλαμβάνει **ένα (1) αρχείο μόνο:** Το αρχείο πηγαίου κώδικα **HashedHeaps.java**, επαρκώς σχολιασμένο.

ΠΡΟΣΟΧΗ στο παραδοτέο σας:

- ***Μην χρησιμοποιήσετε άλλη μορφή συμπίεσης εκτός του .zip.***
- ***Μην τροποποιήσετε το αρχείο HashedHeaps.java, παρά μόνο με τη συμπλήρωση κώδικα προ-διαγεγραμμένων μεθόδων.***
- ***Μην παραδώσετε στο .zip ολόκληρο το project folder στο οποίο εργάζεστε σε IDE της επιλογής σας: ζητείται μόνο το αρχείο HashedHeaps.java.***

ΠΑΡΑΔΟΣΗ αποκλειστικά μέσω της πλατφόρμας «ΑΡΙΣΤΑΡΧΟΣ» του τμήματος, έως και την **14/1/2024, 23:59**. Ανεβάστε το ΑΜ_Επώνυμο_Όνομα.zip στην περιοχή «Εργασίες».

ΕΡΩΤΗΣΕΙΣ/ΑΠΟΡΙΕΣ/ΔΙΕΥΚΡΙΝΙΣΕΙΣ αποκλειστικά μέσα από την Περιοχή Συζήτησης **«Εργασία 2 (2023-2024)»** της πλατφόρμας «ΑΡΙΣΤΑΡΧΟΣ». Δε θα απαντηθούν emails με απορίες.

ΔΕ ΘΑ ΑΞΙΟΛΟΓΗΘΟΥΝ ΕΡΓΑΣΙΕΣ ΠΟΥ: (i) παραδίδονται εκπρόθεσμα ή με άλλο τρόπο (π.χ. email), **(ii)** παραδίδονται σε μορφή συμπίεσης διαφορετική από .zip, **(iii)** περιλαμβάνουν άσχετα αρχεία.

ΒΑΘΜΟΛΟΓΗΣΗ Η εργασία μετρά 20% στον τελικό βαθμό (εφόσον το γραπτό σας βαθμολογηθεί με τουλάχιστον 4). Η εργασία βαθμολογείται με 0, αν δεν δύναται να αξιολογηθεί, π.χ., αν ο κώδικας δεν μεταγλωττίζεται ορθά.

Η εργασία είναι αυστηρά ατομική. *Αντιγραφή επιφέρει άμεσο μηδενισμό και των δύο εργασιών του μαθήματος. Τα παραδοτέα των εργασιών θα ελεγχθούν (όλα ανά ζεύγη).*