Εργασία Τρίτη
"Λεξικό"

Σκοπός της άσκησης

Ο σκοπός της παρούσας άσκησης είναι η υλοποίηση ενός λεξικού με χρήση της δομής του πίνακα κατακερματισμού. Στα πλαίσια της εργασίας θα εξασκηθείτε επίσης στις μεθόδους ταξινόμησης αλφαριθμητικών.

Ερώτημα 1 – Λεξικό (Βάρος 60%)

Ενα λεξικό είναι δυνατόν να υλοποιηθεί με χρήση συνάρτησης κατακερματισμού που εφαρμόζεται στα αλφαριθμητικά-δεδομένα του λεξικού. Η συνάρτηση κατακερματισμού παράγει από το αλφαριθμητικό ένα (μη μοναδικό) αναγνωριστικό που χρησιμοποιείται ως αριθμοδείκτης για να αποθηκευθεί το αλφαριθμητικό στον πίνακα. Αν η θέση του πίνακα χρησιμοποιείται ήδη από κάποιο προηγούμενο αλφαριθμητικό, τότε με βάση τη θεωρία το νέο αλφαριθμητικό εισάγεται σε κάποια άλλη θέση μέσω κατακερματισμού με ανοιχτή διεύθυνση (π.χ. γραμμική διερεύνηση σε κυκλικό πίνακα).

Υλοποιήστε το λεξικό χρησιμοποιώντας το αρχείο με αλφαριθμητικά που σας παρέχεται (dictionary.txt). Το αρχείο περιέχει 183.719 αλφαριθμητικά και ο χρήστης θα επιλέγει πόσα από αυτά θα εισαχθούν στο λεξικό. Για λόγους οικονομίας μνήμης, η κάθε θέση του πίνακα κατακερματισμού θα έχει ως δεδομένα ένα δείκτη προς χαρακτήρα και όχι αλφαριθμητικό σταθερού μήκους. Για κάθε αλφαριθμητικό που διαβάζετε από το αρχείο θα πρέπει να δεσμεύετε με malloc() ακριβώς τις θέσεις μνήμης που χρειάζεστε.

Μετά τη δημιουργία του λεξικού, θα πρέπει να δίνετε στο χρήστη τη δυνατότητα να αναζητήσει κάποιο αλφαριθμητικό - η αναζήτηση θα γίνεται χρησιμοποιώντας το αποτέλεσμα της συνάρτησης κατακερματισμού (ή και γραμμική διερεύνηση). Εάν η λέξη υπάρχει, τότε ο χρήστης ενημερώνεται για (α) την τιμή επιστροφής της συνάρτησης κατακερματισμού και (β) τη θέση του αλφαριθμητικού στον πίνακα.

Αναφορικά με τη συνάρτηση κατακερματισμού, μπορείτε να χρησιμοποιήσετε την απλή παρακάτω συνάρτηση (ή όποια άλλη επιθυμείτε):

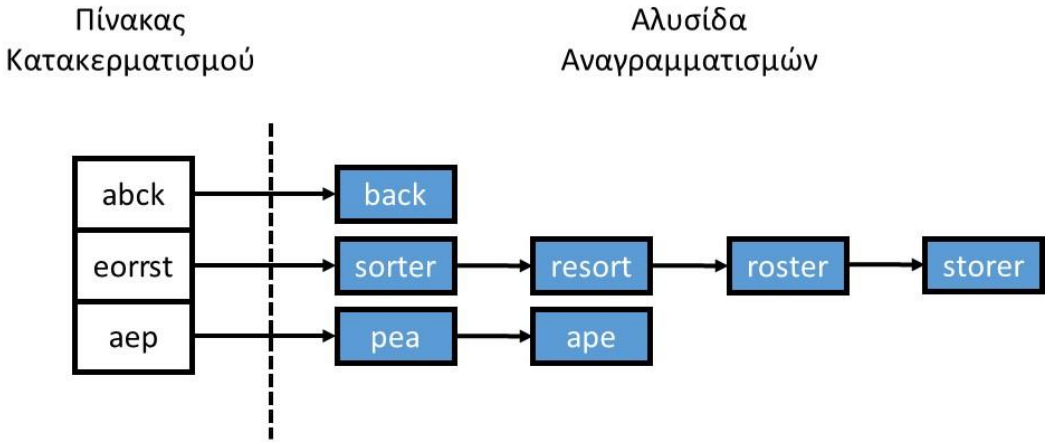
```
unsigned int strtohash(char* str){
     unsigned int hash=0;
     while(*str){
                         hash=*str+31*hash;
                         str++;
     }
     return hash%DICTIONARYSIZE;
}
```

Ερώτημα 2 – Λεξικό με Αναγραμματισμούς (Βάρος 40%)

Δύο αλφαριθμητικά συνιστούν έναν αναγραμματισμό εφόσον περιέχουν ακριβώς τα ίδια γράμματα (π.χ. "pea" και "ape"). Στο λεξικό του προηγούμενου ερωτήματος οι δύο λέξεις θα τοποθετούνταν σε διαφορετικές θέσεις του πίνακα, οπότε είναι δύσκολο να βρεθούν αναγραμματισμοί.

Ενας τρόπος για να παραχθεί το ίδιο αναγνωριστικό για δύο αλφαριθμητικά που συνιστούν αναγραμματισμό είναι να χρησιμοποιηθεί η συνάρτηση κατακερματισμού όχι στα ίδια τα αλφαριθμητικά αλλά στην ταξινόμησή τους. Για το παράδειγμα "pea" και "ape" η ταξινόμηση δίνει το αλφαριθμητικό "aep" το οποίο και παράγει κοινό αναγνωριστικό για τις δύο λέξεις (όπως και για τους συνδυασμους "pae," "eap" και "epa"). Με αυτό τον τρόπο όλοι οι πιθανοί αναγραμματισμοί που υπάρχουν στο λεξικό μπορούν να αντιστοιχιστούν σε κοινά αναγνωριστικά.

Η αποθήκευση των αναγραμματισμών δε θα γίνει με γραμμική διερεύνηση, αλλά με αλυσίδα όπως φαίνεται στο παρακάτω σχήμα.



Για την υλοποίηση της αλυσίδας μπορείτε να τροποποιήσετε τα δεδομένα του πίνακα κατακερματισμού ώστε να έχουν δύο πεδία:

1. Το πρώτο πεδίο είναι το ταξινομημένο αλφαριθμητικό.
2. Το δεύτερο πεδίο είναι ένας δείκτης προς την αλυσίδα που περιέχει τους αναγραμματισμούς. Η αλυσίδα είναι εύκολο να υλοποιηθεί με χρήση απλά συνδεδεμένης λίστας.

Με βάση τα παραπάνω, η εισαγωγή στο λεξικό ακολουθεί τους εξής κανόνες:

1. Αν το αναγνωριστικό δείχνει σε κενή θέση του πίνακα τότε κάνετε εισαγωγή του ταξινομημένου αλφαριθμητικού στον πίνακα και του αταξινόμητου αλφαριθμητικού στην αντίστοιχη αλυσίδα.
2. Αν το αναγνωριστικό δείχνει σε μη κενή θέση του πίνακα και το ταξινομημένο αλφαριθμητικό ταυτίζεται με αυτό στον πίνακα τότε κάνετε εισαγωγή του αταξινόμητου αλφαριθμητικού στην αντίστοιχη αλυσίδα.

3. Αν το αναγνωριστικό δείχνει σε μη κενή θέση του πίνακα και το ταξινομημένο αλφαριθμητικό δεν ταυτίζεται με αυτό στον πίνακα τότε κάνετε γραμμική διερεύνηση μέχρι να καταλήξετε στο βήμα 1 ή 2.

Αντίστοιχα τροποποιείται και η αναζήτηση με βάση το αναγνωριστικό που προκύπτει από το ταξινομημένο αλφαριθμητικό:

1. Αν τα δεδομένα του πίνακα ταυτίζονται με το ταξινομημένο αλφαριθμητικό τότε εκτυπώνετε όλους τους αναγραμματισμούς. Δε χρειάζεται να ελέγχετε εάν το αλφαριθμητικό υπάρχει όντως στην αλυσίδα, αρκεί να βρείτε κάποιον αναγραμματισμό.
2. Αν τα δεδομένα του πίνακα δεν ταυτίζονται με το ταξινομημένο αλφαριθμητικό τότε κάνετε γραμμική διερεύνηση μέχρι να καταλήξετε στο βήμα 1 (επιτυχής αναζήτηση) ή να διερευνήσετε όλο το λεξικό (ανεπιτυχής αναζήτηση).

Παρόμοια με το προηγούμενο ερώτημα, υλοποιήστε το λεξικό χρησιμοποιώντας το αρχείο με αλφαριθμητικά που σας παρέχεται (dictionary.txt). Θα πρέπει να εισάγεται το σύνολο των αλφαριθμητικών (183.719 στο σύνολο και 163.632 χωρίς αναγραμματισμούς). Αναφορικά με την αναζήτηση, ο χρήστης ενημερώνεται για το αν είναι ανεπιτυχής ή επιτυχής, και αν είναι επιτυχής εμφανίζονται (α) η τιμή επιστροφής της συνάρτησης κατακερματισμού, (β) η θέση των αναγραμματισμών στον πίνακα και (γ) όλοι οι διαθέσιμοι αναγραμματισμοί.

Για την ταξινόμηση των αλφαριθμητικών χρησιμοποιείστε μία από τις quicksort ή mergesort.

Παραδοτέα

1. Κώδικας με σχόλια. Ο κώδικας πρέπει να αναφέρει τα μέλη της ομάδας (μέχρι δύο άτομα) και να ανέβει στο e-class μέχρι την ημερομηνία υποβολής. Ο κώδικας θα πρέπει να τρέχει σωστά σε μηχάνημα του Τμήματος (π.χ. Helios, εργαστήριο Dell/Alienware).
2. Αναφορά σε έντυπη μορφή με ενδεικτικές εκτελέσεις του κώδικα που υλοποιήσατε. Η αναφορά δε θα ανέβει στο e-class, αλλά θα την έχετε μαζί σας κατά την εξέταση της εργασίας.

Καλή Επιτυχία