

## Δομές Δεδομένων και Τεχνικές Προγραμματισμού

Διδάσκων: Μανόλης Κουμπάρκης

Υπεύθυνος Εργαστηρίου: Μίλτος Κυριακάκος

### Εργαστήριο 5: Κατακερματισμός (hashing)

Στο εργαστήριο αυτό θα μελετήσουμε την τεχνική του κατακερματισμού και θα χρησιμοποιήσουμε τον σχετικό κώδικα που παρουσιάστηκε στην 14η ενότητα διαλέξεων και βρίσκεται στην ιστοσελίδα του μαθήματος: <http://cgi.di.uoa.gr/~k08/code.html>.

Ο κώδικας που σας έχει δοθεί υλοποιεί **μερικώς** την τεχνική της ανοικτής διευθυνσιοδότησης με διπλό κατακερματισμό (open addressing with double hashing). Τα κλειδιά των στοιχείων που αποθηκεύουμε στον πίνακα κατακερματισμού είναι ακέραιοι.

Ο κώδικας σας αποτελείται από τα αρχεία `HashTableTypes.h`, `HashTableInterface.h`, `menu.h`, `HashTableImplementation.c`, `menu.c` και `main.c`.

Μελετήστε τον κώδικα και κατανοήστε τη λειτουργία του. Δείτε προσεκτικά τις συναρτήσεις κατακερματισμού που χρησιμοποιούνται. Καταλαβαίνετε τη λειτουργία τους; Μεταγλωττίστε τον κώδικα και εκτελέστε τον. Πειραματιστείτε με όλες τις δυνατότητες του μενού λειτουργιών. Για παράδειγμα, να αποθηκεύσετε μερικούς ακραίους στον πίνακα κατακερματισμού και να ψάξετε να βρείτε κάποιους από αυτούς.

Παρατηρείστε ότι τα αρχεία επικεφαλίδας (π.χ., `HashTableTypes.h`) που σας δίνονται περιέχουν την οδηγία `#ifndef`. Πότε χρησιμοποιείται αυτή η οδηγία και γιατί;

Αφού κατανοήσετε πλήρως τον κώδικα που σας δόθηκε, να κάνετε τις ακόλουθες αλλαγές και επεκτάσεις.

1. Υλοποιήστε τη συνάρτηση `p` στο αρχείο `HashImplementation.c` που δίνει τα `probe decrements` και μας βοηθάει να βρίσκουμε τη θέση ενός κλειδιού για το οποίο αρχικά η `h` μας οδηγεί σε σύγκρουση. Έτσι θα έχετε την πλήρη υλοποίηση της τεχνικής της ανοικτής διευθυνσιοδότησης με διπλό κατακερματισμό.
2. Να αλλάξετε τον κώδικα σας ώστε τα στοιχεία που αποθηκεύονται στον πίνακα κατακερματισμού να έχουν κλειδί μια συμβολοσειρά (και όχι ένα ακέραιο όπως είναι στον κώδικα που σας δόθηκε). Για τον πειραματισμό σας, δίνονται τα αρχεία `EnglishWords.txt` που περιέχει λέξεις της Αγγλικής γλώσσας και `Hard.txt` που

περιέχει συμβολοσειρές που είναι σχεδιασμένες ώστε να δυσκολέψουν τους αλγόριθμους σας. Θα χρειαστείτε μια συνάρτηση κατακερματισμού για συμβολοσειρές. Για αρχή, δοκιμάστε τη συνάρτηση η οποία δουλεύει ως εξής:

- a. Προσθέτει τις τιμές των χαρακτήρων της συμβολοσειράς (έτσι παίρνουμε τον κώδικα κατακερματισμού)
- b. Στη συνέχεια επιστρέφει το άθροισμα modulo TABLESIZE (έτσι υπολογίζουμε τη συνάρτηση συμπίεσης)

Θέστε το TABLESIZE ως 41893.

3. Να επεκτείνετε τον κώδικα σας ώστε να μετράει πόσες συγκρούσεις (collisions) γίνονται συνολικά κατά την εισαγωγή στοιχείων στον πίνακα κατακερματισμού για τα δύο δοσμένα αρχεία συμβολοσειρών. Επίσης, ποιος είναι ο μέγιστος και ποιος ο ελάχιστος αριθμός συγκρούσεων που συμβαίνουν, και για ποιο κάδο του πίνακα κατακερματισμού.
4. Να επεκτείνετε τον κώδικα σας ώστε να μετράει το συνολικό χρόνο που πήρε η αποθήκευση ενός αρχείου με συμβολοσειρές καθώς και τον μέσο χρόνο αποθήκευσης για μια λέξη (επιλογή 1 από το μενού λειτουργιών).
5. Συγκρίνετε την παραπάνω συνάρτηση κατακερματισμού με τις άλλες δύο συναρτήσεις κατακερματισμού κατάλληλες για συμβολοσειρές (με χρήση πολυωνυμικών κωδίκων και με χρήση κωδίκων κυκλικής ολίσθησης) που περιέχονται στις διαφάνειες της σχετικής ενότητας. Για να κάνετε τη σύγκριση, υπολογίστε το συνολικό αριθμό συγκρούσεων, τον μέγιστο αριθμό συγκρούσεων, τον ελάχιστο αριθμό συγκρούσεων και το χρόνο αποθήκευσης για κάθε ένα από τα δύο αρχεία. Ποια είναι τα συμπεράσματα σας;