Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα 2020-21

Γρηγόρης Καλλίνικος - 1115201500056
Θεοδόσης Παιδάκης - 1115201500118

Part 1

Οδηγίες εκτέλεσης:

- $ *make* : For complete instructions.
- $ *make r|un*: Runs 'build', then run the resulting executable with default arguments.
- $ *make t|est*: Runs the unit tests.
- $ ./build/main -x [dataset X folder] -w [dataset W file.csv]

  Όλες οι *make*, τρέχουν αυτόματα Valgrind checks.

Δομές / Σχόλια / Παραδοχές:

— Τα Specs αποθηκεύονται σε ένα **Hash Table μεγέθους n**. Πριν διαβάσουμε τα αρχεία, μετράμε τον αριθμό τους και αναλόγως δημιουργούμε το n. Το μέγεθος του πίνακα λοιπόν είναι ο πιο κοντινός prime number, του διπλάσιου πλήθους των Specs.

Για **hashFunction** χρησιμοποιήθηκε η πολυτεσταρισμένη hashCode() συνάρτηση της Java.
Στόχος ήταν όσο το δυνατόν λιγότερα collisions, χωρίς μεγάλο performance hit. Μετά από κάποια tests στο πρόγραμμα μας, φαίνεται να επιτυγχάνεται αρκετά καλά αυτό.
Θεωρούμε επίσης, πως τα πολλά buckets δεν αποτελούν μειονέκτημα όσον αφορά την μνήμη, αφού το καθένα γίνεται allocate μόνο όταν χρειαστεί να αποθηκευτεί πληροφορία εκεί.

Τα **collisions** λύνονται με μια λίστα, στην οποία δείχνει το hashBucket.

— Το κάθε spec από την αρχικοποίηση του συνδέεται με ένα κόμβο κλίκας που αναφέρεται μόνο στον εαυτό του και περιέχει τις πληροφορίες του json αρχείου.

Οι κόμβοι κλίκας θα συνδέονται μεταξύ τους με μια **κυκλική διπλά συνδεδεμένη λίστα**.
Βοηθάει η συγκεκριμένη δομή στο να προσθαφαιρούμε nodes πολύ εύκολα και γρήγορα.

— Με την συγκεκριμένη υλοποίηση, όταν διαβάζουμε ένα ζευγάρι στο dataset W, το μόνο που χρειάζεται να γίνει, είναι να αλλάξουν ορισμένοι pointers των κόμβων της κλίκας, ώστε οι 2 κυκλικές λίστες να ενωθούν.

Όλη η διαδικασία γίνεται σε O(1) χρόνο και χωρίς να χρειάζεται να δημιουργηθεί κάποια έξτρα βοηθητική δομή στην μνήμη(O(1) memory).

— Για το **διάβασμα των αρχείων** των specs χρησιμοποιούνται τα fopen και opendir, μέχρι τα αρχεία json, μετα συνεχίζουμε με το parsing των specs.
Το μετρημα των αρχειων των specs ακολουθει την ιδια λογικη (fopen και opendir).

Για το parsing των αρχείων με τα specs δημιουργήσαμε δικό μας **parser** χρησιμοποιώντας την συνάρτηση strtok της cstring.
Μόλις γίνει parse ένα spec, προστήθεται στο hashtable.

— Για να μην υπάρξουν διπλότυπα στην εκτύπωση των ζευγαριών, μαρκάρεται ο κάθε κόμβος κλίκας την στιγμή που τον εμφανίζουμε.

— Έχουν δημιουργηθεί αρκετά Unit tests που θεωρήσαμε χρήσιμα και γίνεται το απαραίτητο de-allocation στις δομές.