

Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα Project 1

Αλέξιος Σπηλιωτόπουλος - 1115201700147
Φανούρης-Στυλιανός Ψαθόπουλος - 1115201400320

Χειμερινό εξάμηνο 2020-2021

Μεταγλώττιση

Το κυρίως πρόγραμμα μεταγλωττίζεται με την εντολή `make`, στον κατάλογο που βρίσκεται το `makefile`. Για το unit testing γίνεται ξεχωριστή μεταγλώττιση με τις εντολές

- `make test_app`, για το testing συναρτήσεων του κυρίως προγράμματος
- `make test_vector`, για το testing της δομής δυναμικού πίνακα
- `make test_list`, για το testing της δομής λίστας
- `make test_map`, για το testing της δομής του πίνακα κατακερματισμού

Για τον καθαρισμό των αντικειμενικών αρχείων και του εκτελέσιμου, πληκτρολογούμε `make clean`.

Εκτέλεση

Εκτελούμε με την εντολή `./app <data_path> <csv_file_path> [-s <n>]`

- `data_path`, ο κατάλογος των αρχείων δεδομένων
- `csv_file_path`, το μονοπάτι στο αρχείο συσχετήσεων
- `[-s <n>]`, προαιρετικό όρισμα, ορίζει το μέγεθος του πίνακα κατακερματισμού. Αν δεν δοθεί, χρησιμοποιείται ο αριθμός των συνολικών αρχείων δεδομένων ως μέγεθος.

Παραδείγματα εκτέλεσης:

- `./app Datasets Datasets/sigmoid_large_labelled_dataset.csv`
- `./app Datasets Datasets/sigmoid_medium_labelled_dataset.csv -s 137`
- `./app Datasets/camera_specs/2013_camera_specs`
- `./app Datasets/sigmoid_large_labelled_dataset.csv -s 1000`

Για την εκτέλεση των tests, καλούμε το εκτελέσιμο `./test`.

Αρχή του προγράμματος

Μέσω της συνάρτησης `pick_the_buckets`, το πρόγραμμα επεξεργάζεται τα ορίσματα της γραμμής εντολών που δόθηκαν, με σκοπό να βρει αν δόθηκε από τον χρήστη όρισμα για το μέγεθος του πίνακα κατακερματισμού. Υπάρχουν τρεις περιπτώσεις:

- η πρώτη είναι να δόθηκε σωστά ως όρισμα το μέγεθος του πίνακα,
- η δεύτερη είναι να μην δόθηκε κανένα όρισμα για το μέγεθος του πίνακα, επομένως η συνάρτηση υπολογίζει τον συνολικό αριθμό json αρχείων και χρησιμοποιεί τον αριθμό αυτό σαν μέγεθος του hash table.
- η τρίτη περίπτωση είναι να δοθεί λάθος όρισμα ή να μην υπάρχουν αρχεία δεδομένων. Και οι δύο περιπτώσεις θα οδηγήσουν σε σφάλμα και τερματισμό του προγράμματος με κατάλληλο μήνυμα.

JSON Parsing

Στην συνέχεια, μέσω της συνάρτησης `read_data_files`, το πρόγραμμα, διαβάζει όλα τα json αρχεία από όλους τους καταλόγους και αποθηκεύει τις πληροφορίες για κάθε προϊόν στις δομές που αναλύουμε παρακάτω.

Ανάγνωση Σχέσεων Προϊόντων

Η συνάρτηση `read_relations` διαβάζει κάθε γραμμή του csv αρχείου και ενώνει κατάλληλα τις κλίκες των προϊόντων που υπάρχουν σε κάθε γραμμή που υποδηλώνει θετική συσχέτιση. Οι γραμμές που αφορούν αρνητική σχέση αγνοούνται.

Δομές δεδομένων

Έχουν υλοποιηθεί οι εξής δομές:

- Πίνακας κατακερματισμού με αλυσίδες υπερχείλησης
- Δυναμικός πίνακας
- Απλά συνδεδεμένη λίστα (με δείκτη στον τελευταίο κόμβο)

Η χρήση τους θα περιγραφεί παρακάτω.

Πίνακας κατακερματισμού

Οι κόμβοι του πίνακα (`struct map_nodes`) έχουν ως κλειδί μια συμβολοσειρά της μορφής `website//id` και τιμή έναν δείκτη σε κλίκα (`struct clique`). Χρησιμοποιήθηκε πίνακας κατακερματισμού για να έχουμε γρήγορη πρόσβαση στα προϊόντα και τις κλίκες στις οποίες ανήκουν.

Δομή Product

Κάθε προϊόν που γίνεται parsed από ένα json αρχείο, αποθηκεύεται σε ένα struct product με τα εξής μέλη:

- int id, το id του προϊόντος που ταυτίζεται με το όνομα του json αρχείου απ' το οποίο προήλθε,
- char *website, το όνομα του καταλόγου στο οποίο ανήκει το προϊόν,
- struct list *specs, μια λίστα από τα χαρακτηριστικά του αρχείου ως ζεύγη (όνομα, τιμή)¹
- struct product *next, ένα δείκτη σε επόμενο προϊόν μιας δεδομένης κλίκας
- struct clique *clique, ο δείκτης στην κλίκα στην οποία ανήκει το προϊόν

¹ struct specs

Δομή Κλίκας

Η κλίκα έχει υλοποιηθεί με ένα struct clique που έχει τα εξής μέλη:

- int size, όπου αποθηκεύεται ο αριθμός των προϊόντων της κλίκας
- struct product *first, ένας δείκτης στο πρώτο προϊόν της κλίκας,
- struct product *last, ένας δείκτης στο τελευταίο προϊόν της κλίκας

Λειτουργία

Τα προϊόντα που ανήκουν σε μια κλίκα συνδέονται, όπως είδαμε παραπάνω, με έναν δείκτη στο επόμενο προϊόν. Έχοντας άμεση πρόσβαση στο πρώτο και τελευταίο προϊόν της κλίκας, έχουμε κατασκευάσει μια απλά συνδεδεμένη λίστα από προϊόντα.

Κατά την αρχικοποίηση, κάθε προϊόντος, αυτό ανήκει σε μια κλίκα με μοναδικό στοιχείο το συγκεκριμένο προϊόν.

Ανάγνωση θετικής σχέσης και συνένωση κλικών

Όταν διαβάσουμε κάποια θετική σχέση από το csv αρχείο, τότε αφού κάνουμε hash τα προϊόντα με την παραπάνω μορφή, παίρνουμε εφόσον υπάρχουν τα συγκεκριμένα προϊόντα, τους δείκτες στις κλίκες στις οποίες ανήκουν.

Τότε, μέσω του τελευταίου δείκτη σε προϊόν(last_product) της πρώτης κλίκας, ενώνουμε το τελευταίο struct product της πρώτης κλίκας με το πρώτο προϊόν της δεύτερης κλίκας(μέσω του δείκτη της first_product). Στην συνέχεια, για κάθε προϊόν των δύο κλικών, και μέσω του δείκτη σε κλίκα που έχει κάθε struct product, αλλάζουμε κάθε πρώτο προϊόν της εκάστοτε

κλίκας ώστε να δείχνει στο πρώτο προϊόν της πρώτης κλίκας της συσχέτισης, και το τελευταίο προϊόν της εκάστοτε κλίκας ώστε να δείχνει στο τελευταίο προϊόν της δεύτερης κλίκας της συσχέτισης.

Με αυτόν τον τρόπο, καταλήγουμε στη σωστή ενημέρωση κάθε κλίκας.

Δομή Χαρακτηριστικών Προϊόντος

Τα χαρακτηριστικά που διαβάζονται μέσω του parsing των json αρχείων αναπαρίστανται από την δομή struct spec, η οποία έχει τα εξής μέλη:

- char *name, το όνομα του χαρακτηριστικού
- char **value, ένας πίνακας από ονόματα τιμών του χαρακτηριστικού(μπορεί να είναι λίστα από τιμές)
- int cnt, ο αριθμός των παραπάνω τιμών του χαρακτηριστικού

Εκτύπωση αποτελεσμάτων

Με την συνάρτηση print_results, διατρέχουμε το σύνολο του hash table και εκτυπώνουμε όλους τους συνδυασμούς προϊόντων στις διαφορετικές κλίκες. Κλίκες που δείχνουν στο ίδιο πρώτο προϊόν, δεν εκτυπώνονται για να αποφυγεί πολλαπλή εκτύπωση των ίδιων προϊόντων. Η συνάρτηση γυρνάει έναν ακέραιο ώστε να μπορεί να ελεγχθεί εύκολα ο αριθμός των θετικών συσχετίσεων.