

Λειτουργικά Συστήματα

Εργασία 1

Ονοματεπώνυμο: Χριστοδούλου Δημήτρης
Α.Μ. : 1115201500179

Εκτέλεση προγράμματος:

Απλή μέθοδος: `./mygraph (-v)`

Ανεπτυγμένη μέθοδος (για διάβασμα από αρχείο ή/και γράψιμο σε αρχείο):
`./mygraph -i inputFile -o outputFile (-v)`

Το `-v` αναπαριστά `verbose` παράμετρο, η οποία αν υπάρχει ως `command line argument` επιτρέπει περισσότερα `comments` κατά τη λειτουργία του προγράμματος. Δεν ενδείκνυται για εκτέλεση πολλών εντολών από αρχείο, καθώς η ανάγνωση του `output` ενδέχεται να είναι πιο δύσκολη.

Η `verbose` παράμετρος δεν μπορεί να μπει στην περίπτωση που το πρόγραμμα κληθεί με μόνο ένα αρχείο ως παράμετρο.

Δομές και Προγραμματιστικές Παραδοχές:

Η δομή που χρησιμοποιήθηκε για τον γράφο είναι μονά συνδεδεμένη λίστα από λίστες. Η λίστα - “πατέρας” υλοποιεί τους κόμβους του γράφου και οι μικρότερες λίστες αποτελούν τα `transactions` ανάμεσα σε κόμβους.

Συγκεκριμένα, η μεγάλη λίστα έχει κόμβους που περιέχουν το όνομα του κόμβου και το πόσες ακμές φεύγουν από αυτόν. Οι μικρότερες λίστες αποτελούνται από κόμβους που περιέχουν το όνομα του κόμβου που δέχεται την ακμή από τον κόμβο - “πατέρα” και το βάρος της ακμής (ποσό του `transaction`).

Επιπλέον, έχει υλοποιηθεί μια δομή στοίβας με λίστα που χρησιμοποιείται στις λειτουργίες `findCircles` και `circleFind` προκειμένου να κρατάει αποθηκευμένο το μέχρι τώρα προσπελασμένο μονοπάτι.

Τα αρχεία που υπάρχουν στον φάκελο είναι:

- `Makefile`
- `main.c`, περιλαμβάνει το `parsing` των εντολών του χρήστη.
- `graph.c / graph.h`, περιλαμβάνει όλες τις συναρτήσεις που αφορούν τις λειτουργίες του προγράμματος και δευτερεύουσες λειτουργίες σχετικά με τον γράφο.
- `stack.c`, περιλαμβάνει την προαναφερθείσα δομή στοίβας.
- `shared.h`, που περιλαμβάνει έναν `extern` δείκτη σε ένα αρχείο, το οποίο είτε παίρνει την τιμή του `output` αρχείου από τις `command line` παραμέτρους, είτε, αν αυτές δεν δοθούν, παίρνει το `stdout`.

Πολυπλοκότητες των Εντολών:

Στα παρακάτω θεωρούμε n το πλήθος των κόμβων του γράφου και m το πλήθος των ακμών του εκάστοτε κόμβου.

- Insert, Delete (Node): Οι 2 αυτές εντολές έχουν πολυπλοκότητα $O(n)$, καθώς πρέπει να διατρέξουν τη λίστα.
- Insert, Delete (Edge): Οι 2 αυτές εντολές έχουν πολυπλοκότητα $O(n + m)$, καθώς πρέπει να διατρέξουν τη λίστα των κόμβων μέχρι να βρουν τον κόμβο που ζητείται και μετά να διατρέξουν τη λίστα των ακμών του.
- Receiving: Η εντολή έχει πολυπλοκότητα $O(n*m)$, αφού πρέπει να διατρέξουμε όλη την λίστα και τις υπο-λิสτες.
- circleFind: $O(n \log(n))$ στην μέση περίπτωση, καθώς διατρέχουμε όλους τους κόμβους και τις ακμές και για κάθε ακμή ελέγχουμε τους δικούς της γείτονες. Το $\log(n)$ προκύπτει ως το μέσο μήκος κάθε διαδρομής στο γράφο.
- findCircles: $O(n \log(n))$, καθώς η συνάρτηση αυτή είναι παρόμοια με τη circleFind.
- traceFlow: $O(n^4)$ στην χειρότερη περίπτωση, όπου l το μέγιστο μήκος της διαδρομής