

Ανάπτυξη Λογισμικού Πληροφορικής

Χειμερινό Εξάμηνο 2014 - 2015



“Σύστημα Ανάλυσης Κοινωνικών δικτύων”

Υπεύθυνος καθηγητής

Γιάννης Ιωαννίδης

Συνεργάτες μαθήματος

Μάνος Καρβούνης
Μαριαλένα Κυριακίδη

Χρήστος Μάλλιος
Αντωνία Σαραβάνου
Χριστόφορος Σβίγγος
Σταμάτης Χριστοφορίδης
Γιάννης Χρόνης

ΠΕΡΙΕΧΟΜΕΝΑ

Γενική περιγραφή

Περιγραφή παραδοτέων μαθήματος

1. Υλοποίηση δομών δεδομένων και βασικών λειτουργιών σε γράφο

Γενική περιγραφή

Το θέμα της φετινής εργασίας είναι μια παραλλαγή του θέματος του περσινού προγραμματιστικού διαγωνισμού του Sigmod: η υλοποίηση ενός συστήματος ανάλυσης κοινωνικών δικτύων.

Ένα κοινωνικό δίκτυο απεικονίζεται ως ένας γράφος με κόμβους, τις οντότητες που σχετίζονται μεταξύ τους (άνθρωποι, εταιρίες, κτλ.), και ακμές τις μεταξύ τους σχέσεις (γνωρίζει, εμπιστεύεται, κτλ.). Ένα σύστημα ανάλυσης ενός κοινωνικού δικτύου χρησιμοποιεί μία σειρά μεθόδων για να αναλύσει τη δομή του και να αναδείξει πρότυπα που προκύπτουν από τις συσχετίσεις μεταξύ των οντοτήτων του.



Στα πλαίσια αυτής της εργασίας θέλουμε να αναπτύξετε μια εφαρμογή η οποία θα α) επεξεργάζεται δεδομένα που προέρχονται από ανθρώπους που δουλεύουν σε εταιρίες/πανεπιστήμια και που συμμετέχουν (γράφοντας ποστ) σε φόρουμ διάφορων θεμάτων και β) θα απαντάει σε ερωτήματα σχετικά με τα δεδομένα αυτά.

Το dataset που θα χρησιμοποιήσετε είναι το ίδιο με αυτό που δόθηκε στον προγραμματιστικό διαγωνισμό Sigmod 2014 και μπορείτε να το βρείτε στη σελίδα του διαγωνισμού στον ακόλουθο σύνδεσμο: <http://www.cs.albany.edu/~sigmod14contest/task.html>.

Περιγραφή παραδοτέων μαθήματος

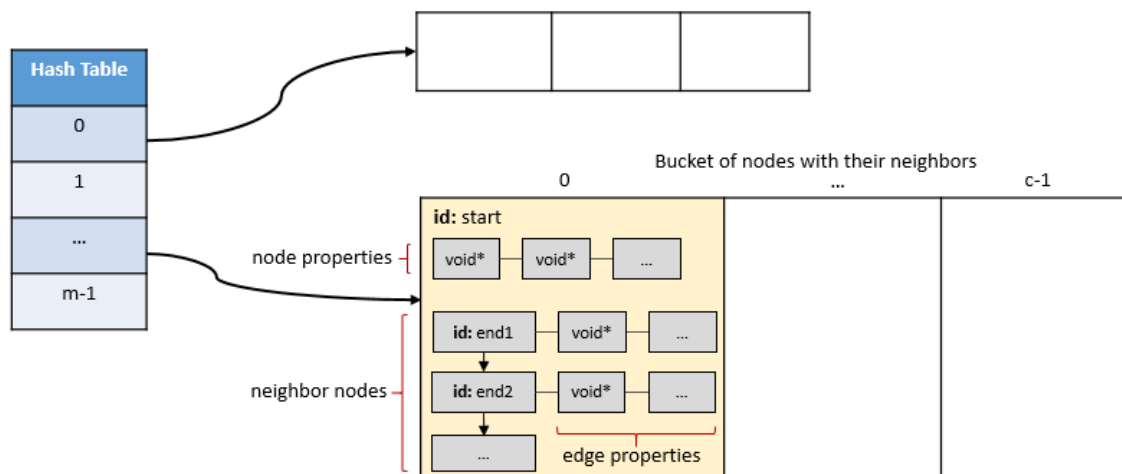
Η εργασία θα χωριστεί σε τέσσερα επίπεδα καθένα με τη δική του ημερομηνία παράδοσης. Δώστε ιδιαίτερη προσοχή στον τρόπο υλοποίησης των επιμέρους επιπέδων καθώς κάθε νέο επίπεδο θα χρησιμοποιεί τις υλοποιήσεις που φτιάξετε στα προηγούμενα. Ακολουθεί αναλυτική περιγραφή των επιμέρους επιπέδων.

1. Υλοποίηση δομών δεδομένων και βασικών λειτουργιών σε γράφο.

Στο επίπεδο αυτό θα υλοποιήσετε τη δομή που θα έχετε για να αποθηκεύετε τα δεδομένα σας σε μορφή γράφου στη μνήμη. Η δομή αυτή θα είναι ένας πίνακας γραμμικού κατακερματισμού (linear hashing). Την περιγραφή του αλγορίθμου μπορείτε να τη βρείτε στον σύνδεσμο: http://cgi.di.uoa.gr/~ad/MDE515/e_ds_linearhashing.pdf.

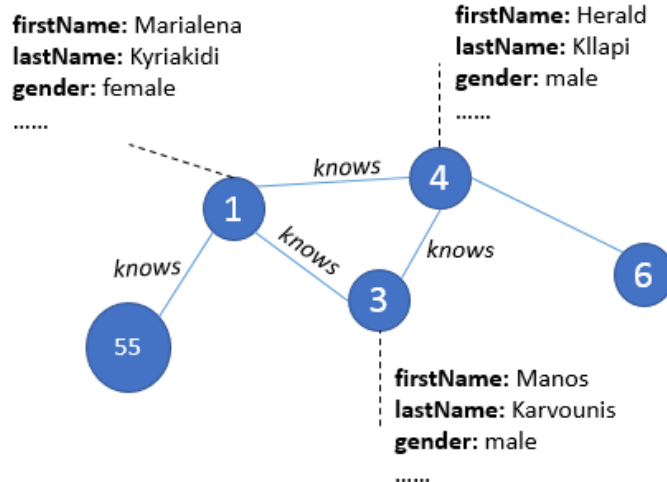
Σύμφωνα με τον παραπάνω αλγόριθμο, αν θεωρήσουμε ότι είμαστε στο i -οστό επίπεδο δημιουργίας split, έχουμε τα παρακάτω δεδομένα και τη γενικότερη σχηματική αναπαράστασή τους.

- hashtable αρχικού μεγέθους m
- c πλήθος κελιών σε κάθε κουβά
- hash function στο i -οστό επίπεδο δημιουργίας split $h_i(k) = k\%(2^i \cdot m)$ όπου k είναι το κλειδί (πάντα ακέραιος).

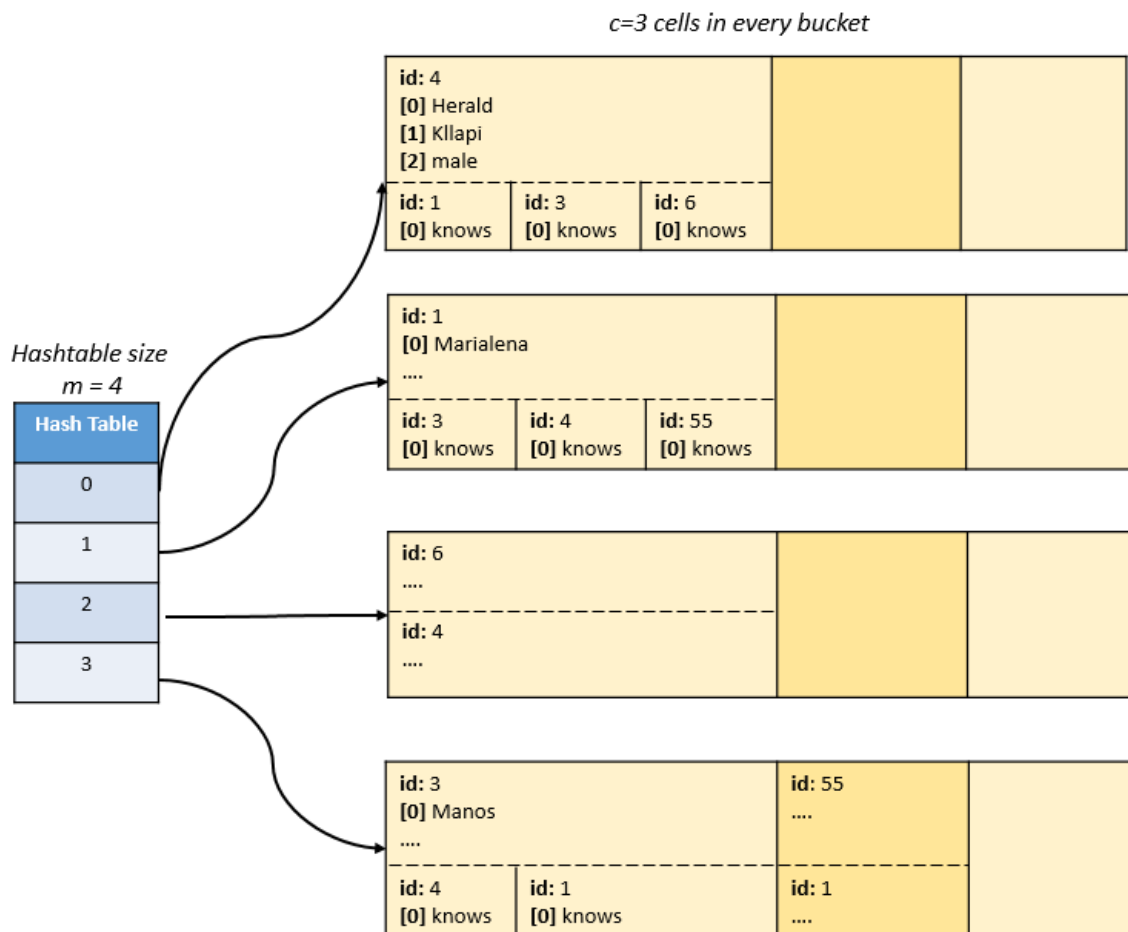


Σχήμα 1. Εσωτερική αναπαράσταση γράφου στο σύστημα

Στα παρακάτω σχήματα ακολουθεί ένα μικρό παράδειγμα κοινωνικού δικτύου μαζί με την εσωτερική αναπαράστασή του, στο επίπεδο δημιουργίας split 0.



Σχήμα 2. Δείγμα κοινωνικού δικτύου



Σχήμα 3. Εσωτερική αναπαράσταση παραδείγματος

Στα πλαίσια αυτής της περιγραφής θα υλοποιήσετε μεθόδους:

- Δημιουργίας γράφου (**createGraph**).

Η συνάρτηση αυτή δημιουργεί ένα άδειο γράφο.

Input: έναν ακέραιο m που είναι το αρχικό μέγεθος του hashtable, έναν ακέραιο c που είναι το πλήθος των κελιών κάθε κουβά.

Output: δείκτη στον άδειο γράφο.

- b. καταστροφής του γράφου (**destroyGraph**).

Η συνάρτηση αυτή καταστρέφει το γράφο.

Input: δείκτη στο γράφο.

Output: success/fail

- c. Εισαγωγής κόμβου στο γράφο (**insertNode**)^{**1}

Η συνάρτηση αυτή εισάγει ένα κόμβο με τα χαρακτηριστικά του στο γράφο.

Input: δείκτη σε δομή με τον κόμβο και τις ιδιότητες του, δείκτη στον γράφο.

Output: success/fail.

- d. Εισαγωγής ακμής στο γράφο (**insertEdge**)

Η συνάρτηση αυτή εισάγει μία ακμή με τις ιδιότητές της στη λίστα γειτνίασης ενός κόμβου.

Input: το id του αρχικού κόμβου, δείκτη στην ακμή και τις ιδιότητες της, δείκτη στον γράφο.

Output: success/fail.

- e. Αναζήτησης ενός κόμβου (**lookupNode**)

Η συνάρτηση αυτή θα επιστρέφει ένα δείκτη στο ζητούμενο κόμβο, δίνοντας πρόσβαση σε ό,τι τον αφορά (ιδιότητες κόμβου και λίστα γειτνίασης με τις ακμές και τις ιδιότητές τους).

Input: το id του ζητούμενου κόμβου, δείκτη στο γράφο.

Output: δείκτη στον κόμβο.

- f. Εύρεσης συντομότερου μονοπατιού σε μη κατευθυνόμενο γράφημα χωρίς βάρη.

- i. από έναν αρχικό κόμβο σε έναν τελικό (**reachNode1**).

Input: id αρχικού κόμβου, id τελικού κόμβου, δείκτη στο γράφο.

Output: έναν ακέραιο με την απόσταση.

- ii. από έναν αρχικό κόμβο σε όλους (**reachNodesN**)

Input: id αρχικού κόμβου, δείκτη στο γράφο.

Output: ένα ResultSet^{**2} από (id, απόσταση).

Σημειώσεις

- (^{**1}) Οι κόμβοι μέσα σε κάθε κουβά πρέπει να είναι ταξινομημένοι για να μπορέσετε να χρησιμοποιήσετε αλγόριθμο αναζήτησης μικρότερης πολυπλοκότητας από τη σειριακή.
- (^{**2}) Ένα ResultSet είναι μια δομή που εμπεριέχει ένα σύνολο τιμών. Αντί όλες οι τιμές να είναι διαθέσιμες εξ αρχής, π.χ. σε έναν πίνακα, το ResultSet δίνει τη δυνατότητα να πάρουμε τις τιμές μία-μία καλώντας διαδοχικά μια συνάρτηση next. Εσωτερικά η δομή ResultSet κρατάει πληροφορίες ώστε να γνωρίζει την τρέχουσα θέση της μέσα στο σύνολο τιμών και πώς μπορεί να ανακτήσει την επόμενη διαθέσιμη τιμή.