



Project Phase II

ΗΥ-240 – Δομές Δεδομένων

Διδάσκουσα: Παναγιώτα Φατούρου

Ιάκωβος Γ. Κολοκάσης

21 Απριλίου 2018

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

1. Εισαγωγή
2. Περιγραφή Δομών
3. Δομές Χρηστών
4. Πίνακας Κατακερματισμού Tweets
5. Γεγονότα
6. Είδη Γεγονότων

Εισαγωγή

- Ιστοσελίδα Μαθήματος: <http://www.csd.uoc.gr/~hy240b>
- Τρόπος Παράδοσης: Με το πρόγραμμα **turnin**
- Ημερομηνία Παράδοση: Δευτέρα 14 Μαΐου 2018, ώρα 23:59
- Διαθέσιμες Υλοποιήσεις: C/C++ / JAVA
- Ερωτήσεις/Διευκρινήσεις: hy240b-list@csd.uoc.gr
- **ΠΡΟΣΟΧΗ:** Compile / Run στα μηχανήματα της σχολής

Αντίστοιχα με την 1η Φάση έχουμε:

- Χρήστες που στέλνουν και διαβάζουν σύντομα μηνύματα, τα οποία ονομάζονται τουίτς (tweets)
- Ο κάθε χρήστης μπορεί να ακολουθεί (follow) άλλους χρήστες με σκοπό να εμφανίζονται τα tweets τους στην περιοχή του (Wall)

Έχουμε λοιπόν δύο βασικές οντότητες:

1. Χρήστες
2. Tweets

Περιγραφή Δομών

Οι γνώσεις που θα χρειαστούν για την υλοποίηση της 2ης φάσης της εργασίας, μεταξύ άλλων, είναι οι εξής:

- **Δυναδικά Δέντρα Αναζήτησης (Δ.Δ.Α)**
 - Αναζήτηση, Εισαγωγή, Διαγραφή
 - Διάσχιση ταξινομημένου Δ.Δ.Α
 - Κόμβοι φρουροί
- **Εμπλουτισμένα Φυλλοπροσανατολισμένα Δ.Δ.Α**
 - Αναζήτηση, Εισαγωγή, Διαγραφή, Διάσχιση
- **Πίνακες Κατακερματισμού**
 - Τεχνική Διπλού κατακερματισμού με ανοικτή διευθυνσιοδότηση (double hashing) (για την επίληση συγκρούσεων)

Η εργασία αποτελείται από 2 ομάδες δομών:

1. Δομές που αφορούν τους χρήστες
2. Δομές που αφορούν τα tweets

Δομές Χρηστών

Οι δομές χρηστών οργανώνονται ως εξής:

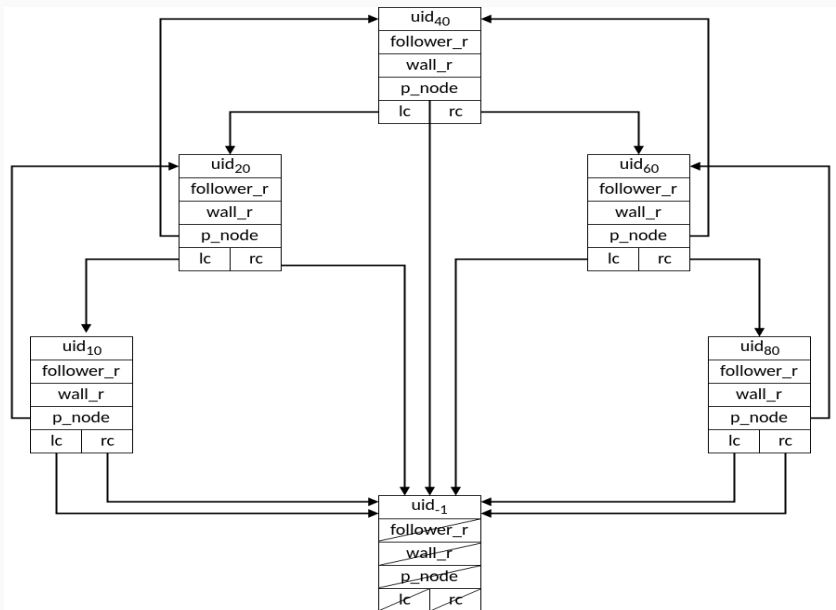
- Οι χρήστες αποθηκεύονται σε ένα διπλά συνδεδεμένο δυαδικό δένδρο αναζήτησης, με κόμβο φρουρό ταξινομημένο με βάση το αναγνωριστικό του χρήστη
- Ο κάθε χρήστης είναι ένα αντικείμενο τύπου `struct Tree_user`

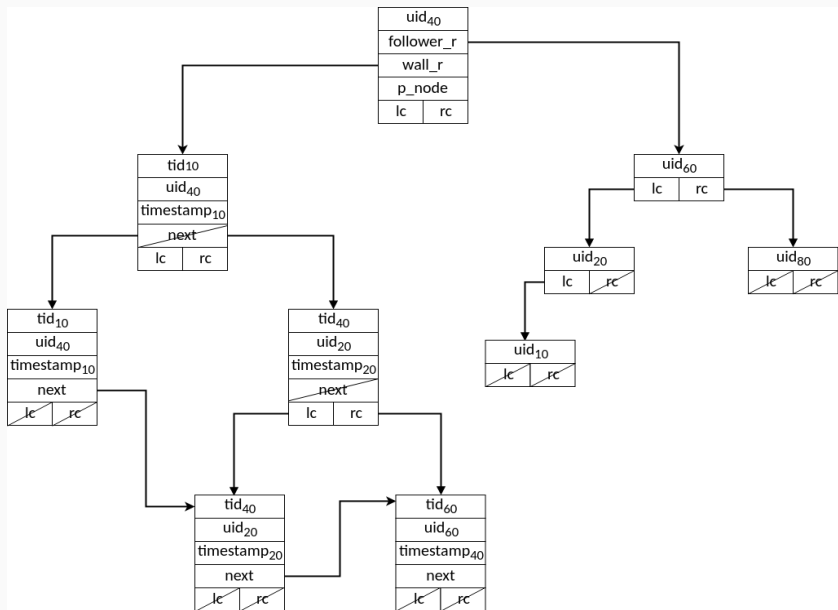
- Κάθε χρήστης περιέχει ένα δείκτη που δείχνει στον κόμβο ρίζα ενός απλού δυαδικού δένδρου αναζήτησης, που ονομάζεται δένδρο οπαδών (followers)
- Κάθε χρήστης περιέχει ένα δείκτη που δείχνει στον κόμβο ρίζα ενός εμπλουτισμένου φυλλοπροσανατολισμένου δένδρου δυαδικής αναζήτησης, που ονομάζεται δένδρο των tweets (Wall)

- uid** Αναγνωριστικό χρήστη (int)
- followers_R** Δείκτης που δείχνει στον κόμβο ρίζα ενός απλού δυαδικού δένδρου αναζήτησης (struct follower *)
- wall_R** Δείκτης που δείχνει στον κόμβο ρίζα ενός εμπλουτισμένου φυλλοπροσανατολισμένου δένδρου δυαδικής αναζήτησης (struct tweet *)
- p_node** Δείκτης προς τον γονικό κόμβο (struct user*)
 - lc** Δείκτης στο αριστερό θυγατρικό κόμβο (struct user*)
 - rc** Δείκτης στο δεξί θυγατρικό κόμβο (struct user*)

- `uid` Αναγνωριστικό χρήστη / οπαδού (int)
- `lc` Δείκτης στο αριστερό θυγατρικό κόμβο (struct follower*)
- `rc` Δείκτης στο δεξί θυγατρικό κόμβο (struct follower*)

- tid** Αναγνωριστικό tweet (int)
- uid** Αναγνωριστικό χρήστη που δημοσίευσε το tweet (int)
- timestamp** Ακέραιος που αντιστοιχεί στην ημερομηνία δημοσίευσης του tweet (int)
- lc** Δείκτης στο αριστερό θυγατρικό κόμβο (struct tweet*)
- rc** Δείκτης στο δεξί θυγατρικό κόμβο (struct tweet*)
- next** Δείκτης προς το επόμενο φύλλο (struct tweet*)





Πίνακας Κατακερματισμού Tweets

Ο πίνακας κατακερματισμού των tweets οργανώνεται ως εξής:

- Αποθηκεύει όλα τα tweets που υπάρχουν στο σύστημα
- Η επίλυση των συγκρούσεων γίνεται βάσει της τεχνικής του διπλού κατακερματισμού με ανοικτή διευθυνσιοδότηση
- Οι συναρτήσεις κατακερματισμού που θα χρησιμοποιήσετε είναι:

Hash Functions

$$h_1(x) = x \% \text{size_of_hashTable}$$

$$h_2(x) = x^2 \% \text{size_of_hashTable}$$

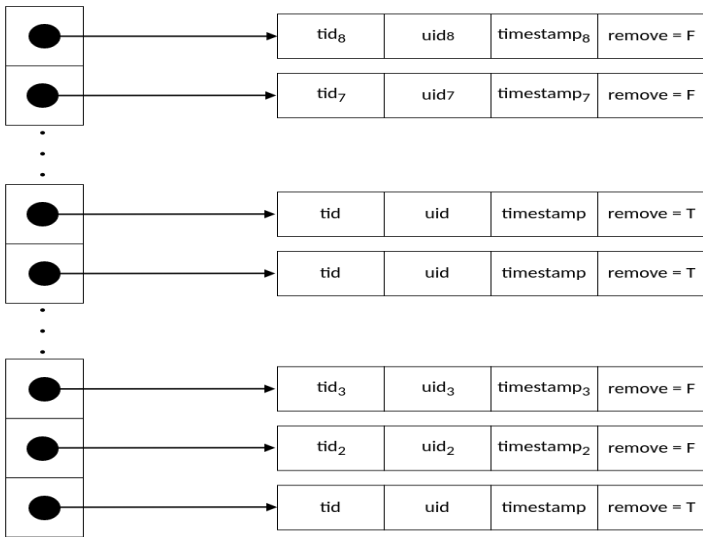
- Ο μέγιστος πλήθος tweets που μπορεί να υπάρχουν στο σύστημα είναι 1000

Πίνακας Κατακερματισμού Tweets

tid	Αναγνωριστικό tweet (int)
uid	Αναγνωριστικό χρήστη που δημοσίευσε το tweet (int)
timestamp	Ακέραιος που αντιστοιχεί στην ημερομηνία δημοσίευσης του tweet (int)
remove	Ένα bit το οποίο υποδηλώνει αν το tweet έχει διαγραφεί ή όχι (char)

Hash Table

Entries



Γεγονότα

Ένα γεγονός αποτελεί μία ενέργεια που εκτελείται μέσα στο σύστημά μας. Για παράδειγμα :

- Εισαγωγή ενός χρήστη
- Ανάρτηση ενός τweet
- κ.τ.λ.

- Τα γεγονότα αυτά είναι καταχωρημένα σε κάποια αρχεία (test files)
- Τα αρχεία υπάρχουν στη σελίδα του μαθήματος
- Το πρόγραμμα κάνει parse αυτά τα αρχεία και εκτελεί τη συνάρτηση που αντιστοιχεί στο εκάστοτε γεγονός
- Δεν χρειάζεται να γραφτεί επιπλέον κώδικας για το χειρισμό των γεγονότων
- Το μόνο που απουσιάζει είναι η υλοποίηση των συναρτήσεων

Ένα γεγονός έχει την εξής μορφή:

```
event_id [arithmetic_arguments [...]]
```

Είδη Γεγονότων

Εισαγωγή ενός νέου χρήστη στο σύστημα

$R \text{ } uid$

Ορίσματα:

uid Αναγνωριστικό του νέου χρήστη

Κατά το γεγονός αυτό:

1. Θα γίνει εισαγωγή ενός νέου χρήστη με αναγνωριστικό `[uid]` στο δένδρο χρηστών
2. **Προσοχή:** Μετά από κάθε εισαγωγή, το δένδρο χρηστών πρέπει να παραμένει ταξινομημένο

Εγγραφή του χρήστη με αναγνωριστικό $[uid_2]$ στο δένδρο των followers του χρήστη με αναγνωριστικό $[uid_1]$

Subscribe

$S \ uid_1 \ uid_2$

Ορίσματα :

uid_1 Αναγνωριστικό του χρήστη που έλαβε ένα νέο follower

uid_2 Αναγνωριστικό χρήστη που κάνει follow

Κατά το γεγονός αυτό:

1. Αναζητούμε στο δένδρο χρηστών, το χρήστη με αναγνωριστικό uid_1
2. Εισάγουμε στο δένδρο των followers ένα νέο κόμβο με αναγνωριστικό uid_2
3. **Προσοχή:** Μετά από κάθε εισαγωγή, το δένδρο των followers θα πρέπει να παραμένει ταξινομημένο

Δημοσίευση ενός tweet με αναγνωριστικό *tid* από τον χρήστη με αναγνωριστικό *uid*

Tweet

T uid tid timestamp

Ορίσματα :

uid Αναγνωριστικό του χρήστη που δημοσίευσε το tweet

tid Αναγνωριστικό του tweet

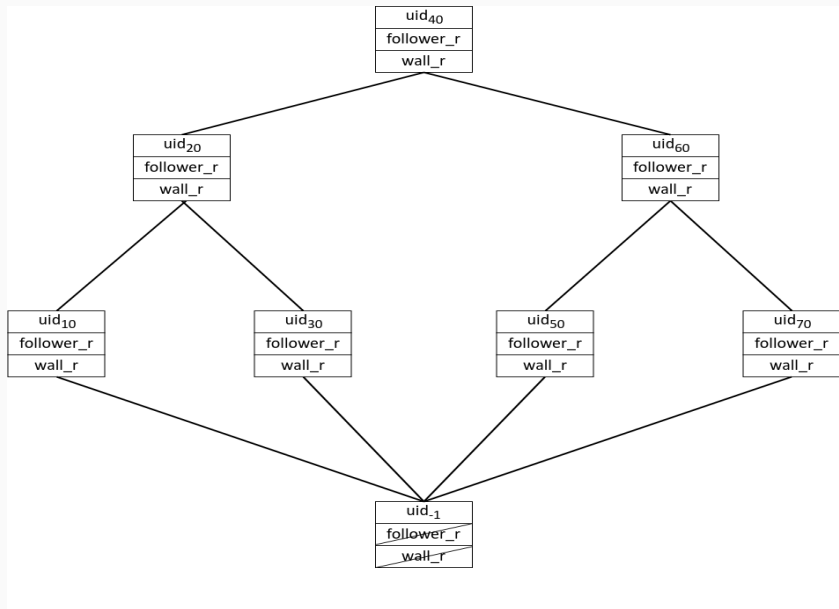
timestamp Χρονική στιγμή στην οποία δημοσιεύτηκε το tweet

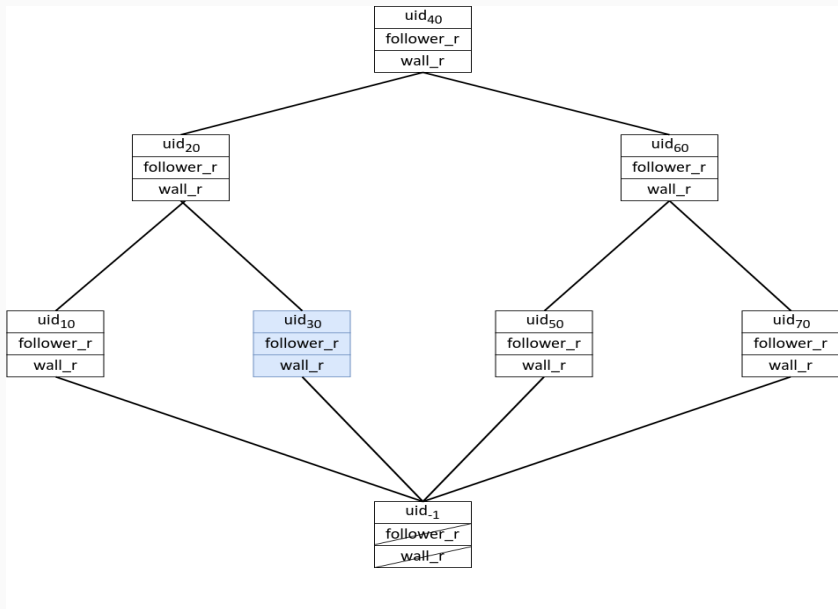
Κατά το γεγονός αυτό:

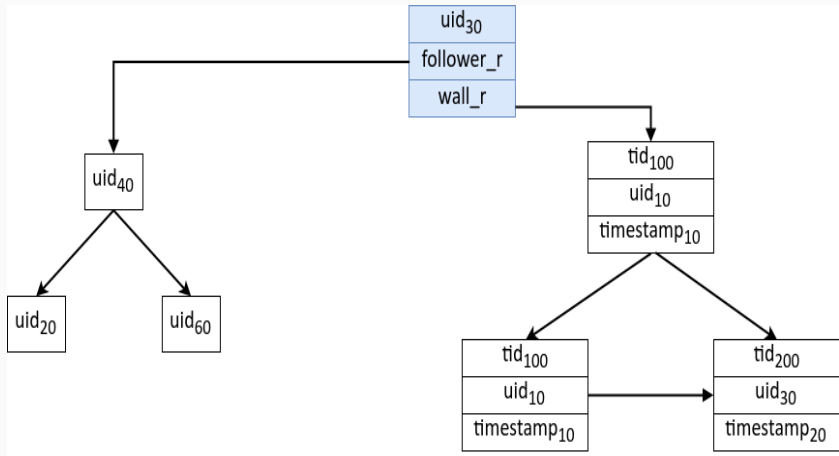
1. Αναζητάμε στο δένδρο χρηστών το χρήστη με αναγνωριστικό *uid*
2. Εισάγουμε στο δένδρο των tweets του, ένα νέο κόμβο με αναγνωριστικό *tid*
3. Διατρέχουμε το δένδρο των followers του χρήστη αυτού και για κάθε κόμβο στο δένδρο:
 - 3.1 Αναζητούμε το χρήστη με το τρέχον *uid* στο δένδρο χρηστών
 - 3.2 Αφού τον εντοπίσουμε, εισάγουμε στη δένδρο των tweets του, ένα νέο κόμβο με αναγνωριστικό *tid*
 - 3.3 **Προσοχή:** Η διαδικασία αυτή θα πρέπει να ολοκληρώνεται σε χρόνο $O(n)$, όπου n το πλήθος των κόμβων στο δένδρο χρηστών
4. Τέλος, εισάγουμε και μια νέα εγγραφή στον πίνακα κατακερματισμού των tweets

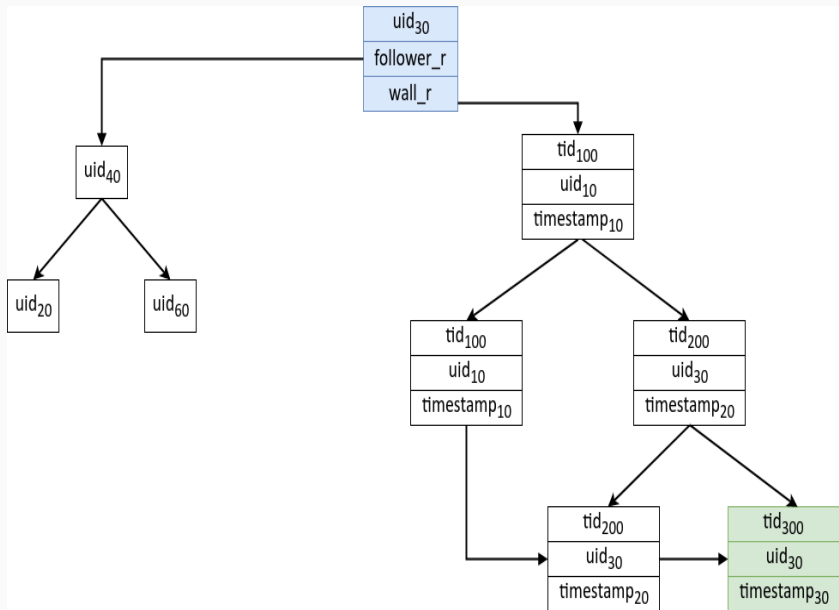
Tweet

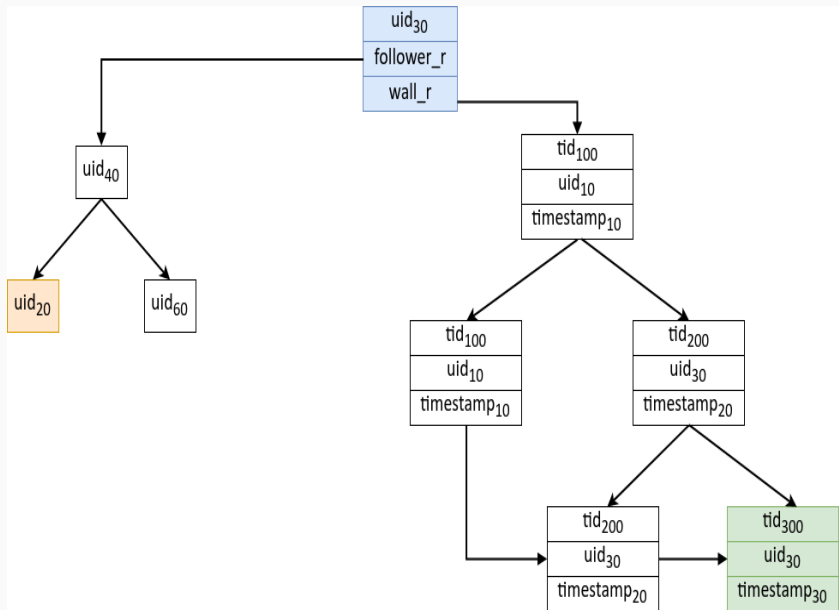
T 30 300 20180420

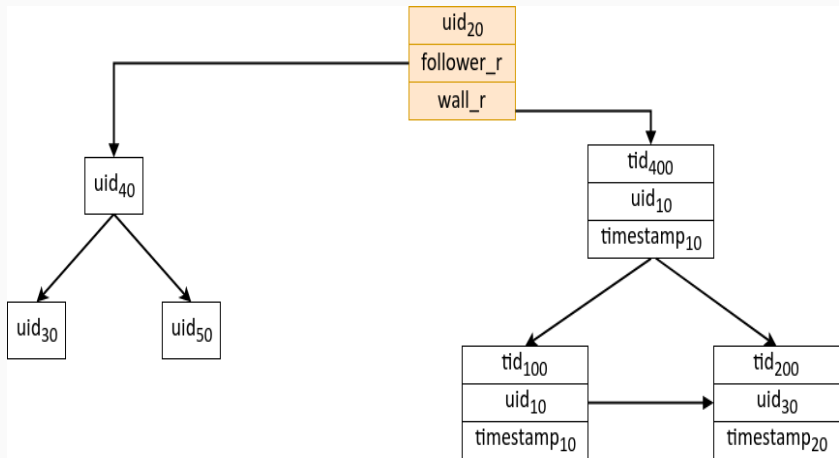


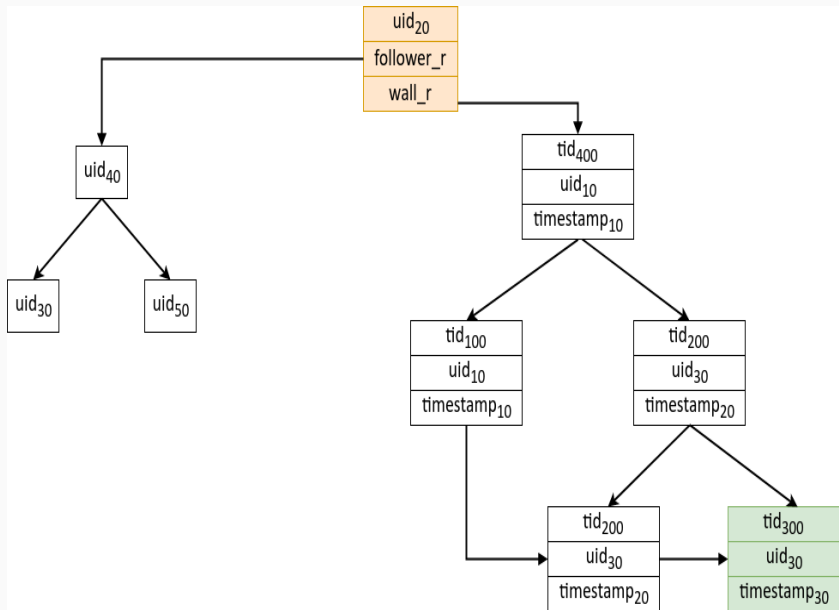


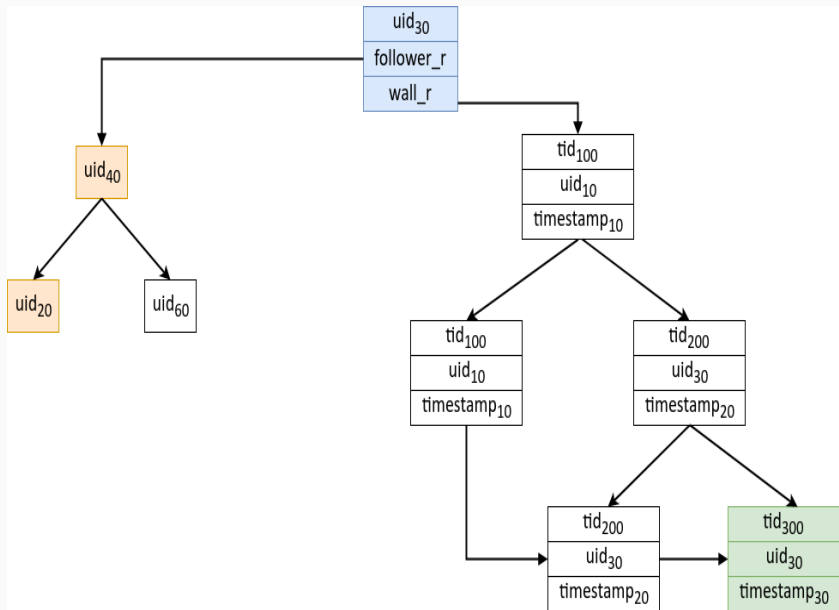


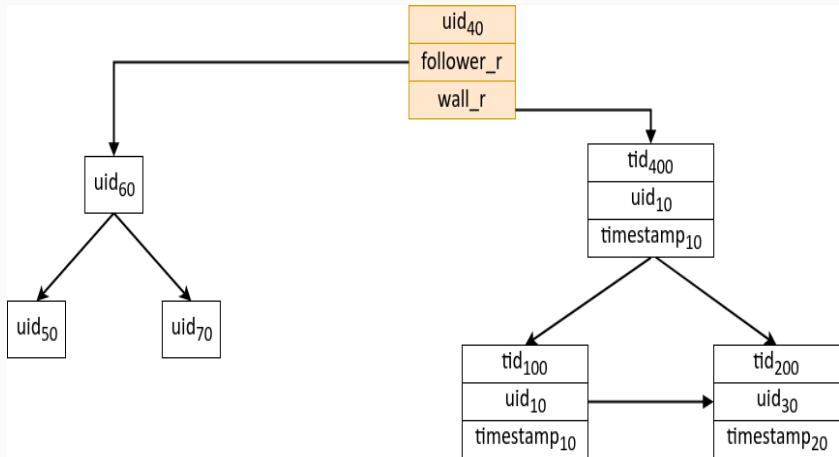


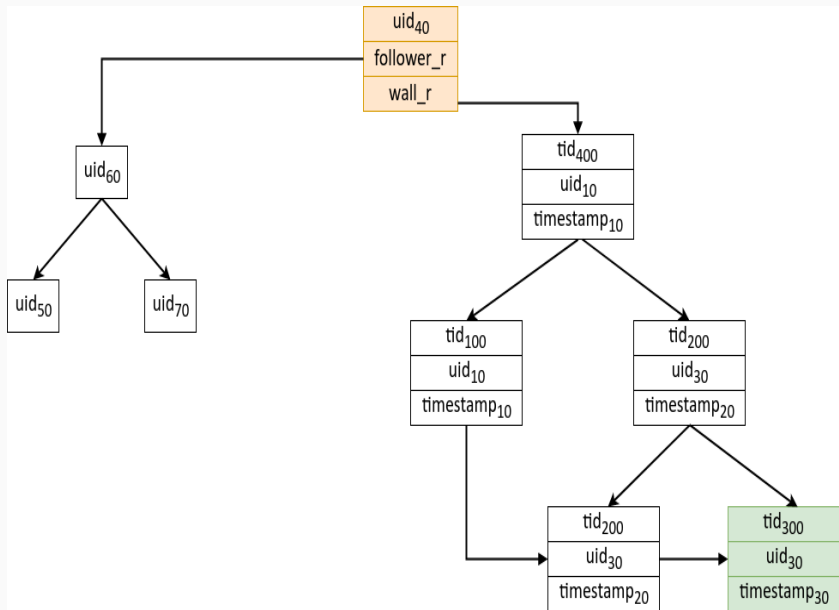


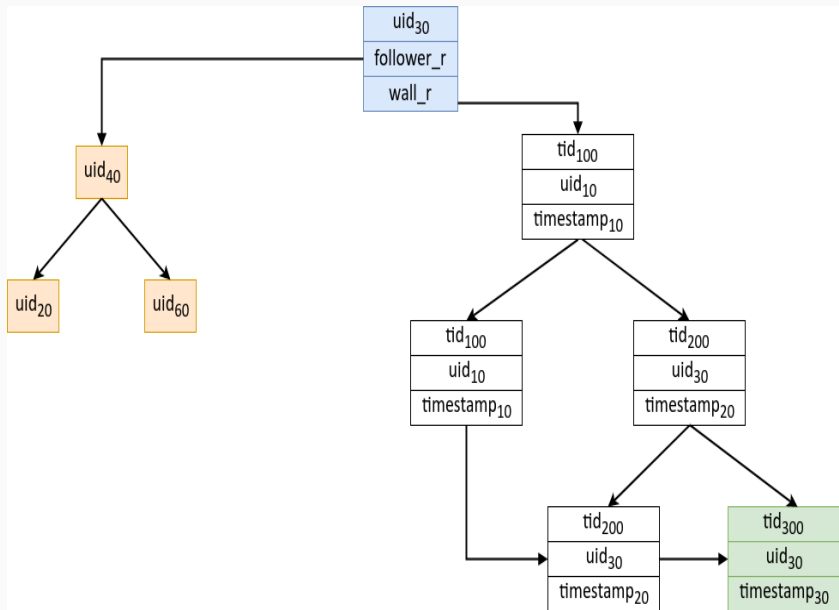


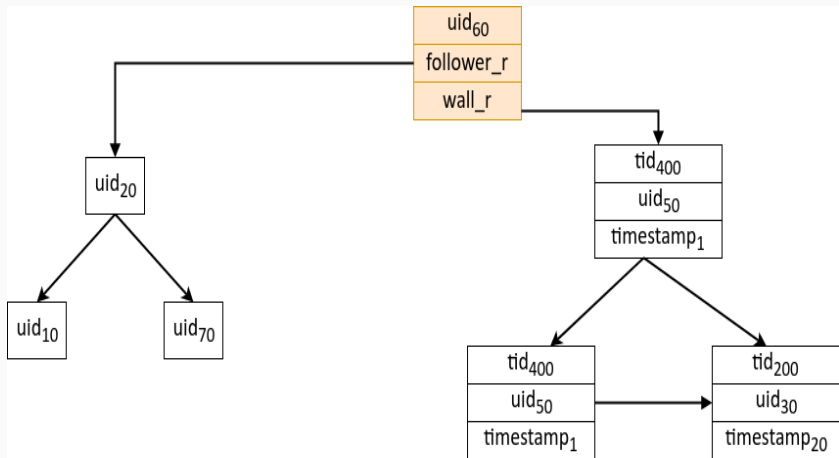


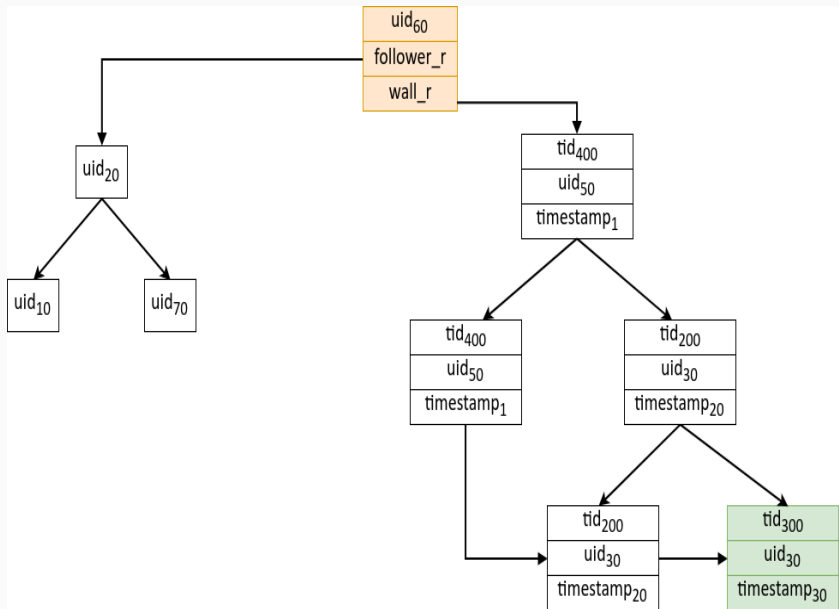












Διαγραφή του χρήστη με αναγνωριστικό uid_2 από τη λίστα των followers του χρήστη με αναγνωριστικό uid_1

Unsubscribe

$U\ uid_1\ uid_2$

Ορίσματα :

uid_1 Αναγνωριστικό του χρήστη που έχασε ένα follower

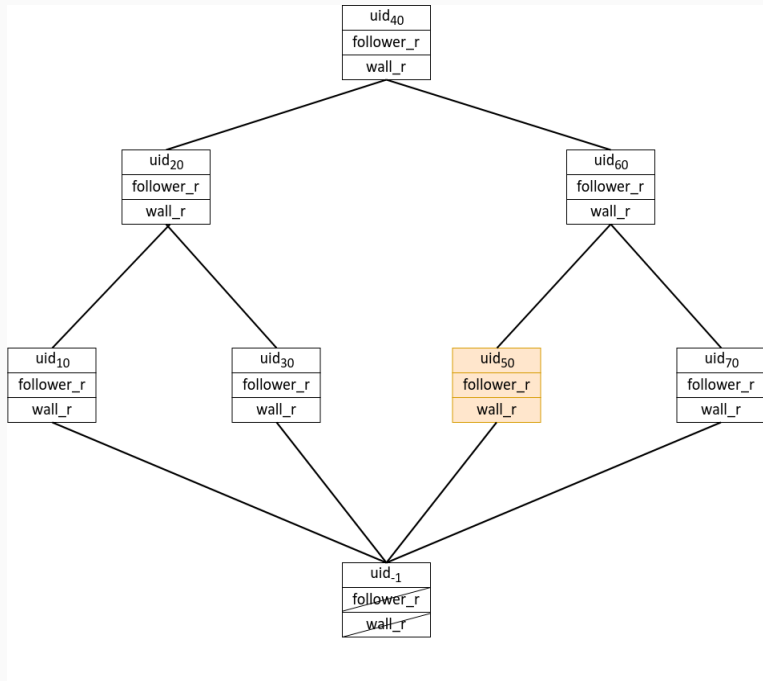
uid_2 Αναγνωριστικό του χρήστη που κάνει unsubscribe

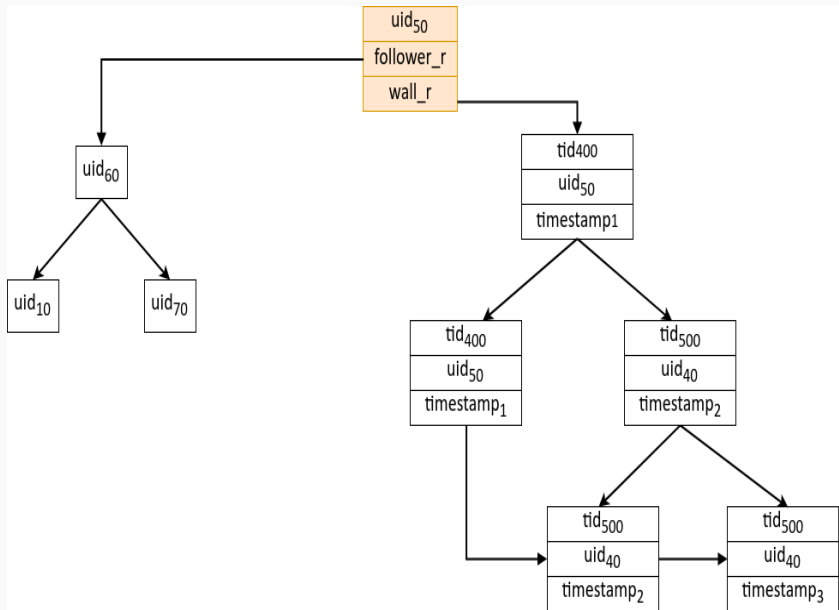
Κατά το γεγονός αυτό:

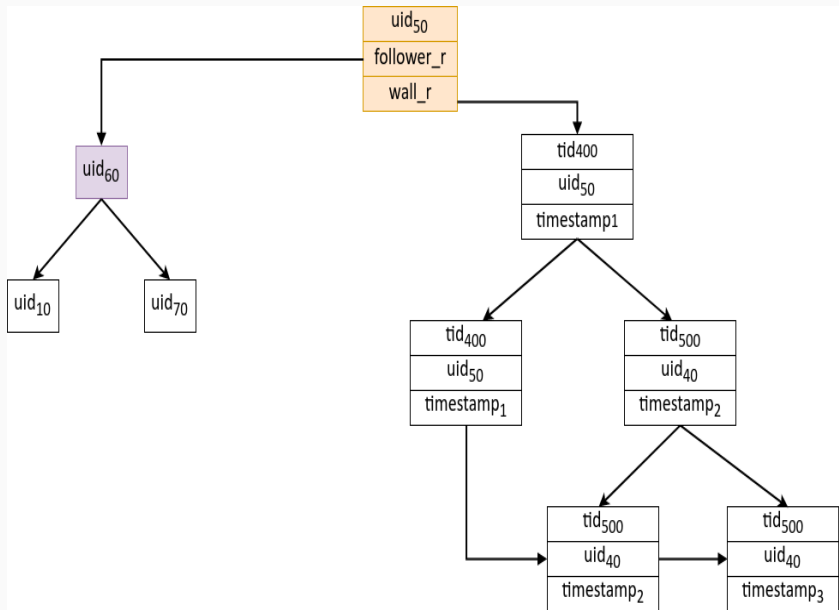
1. Αναζητούμε στο δένδρο χρηστών το χρήστη με αναγνωριστικό uid_1
2. Αφαιρούμε από το δένδρο των followers τον κόμβο με αναγνωριστικό uid_2
3. Αναζητούμε στο δένδρο χρηστών το χρήστη με αναγνωριστικό uid_2
4. Διατρέχουμε το δένδρο των tweets του χρήστη με αναγνωριστικό uid_2
5. Αφαιρούμε από αυτή τους κόμβους που έχουν το πεδίο uid ίσο με uid_1

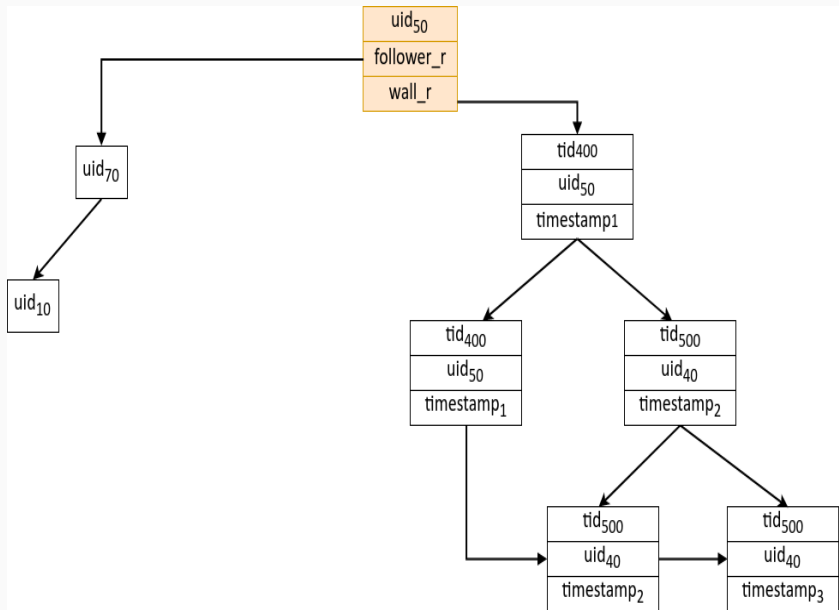
Unsubscribe

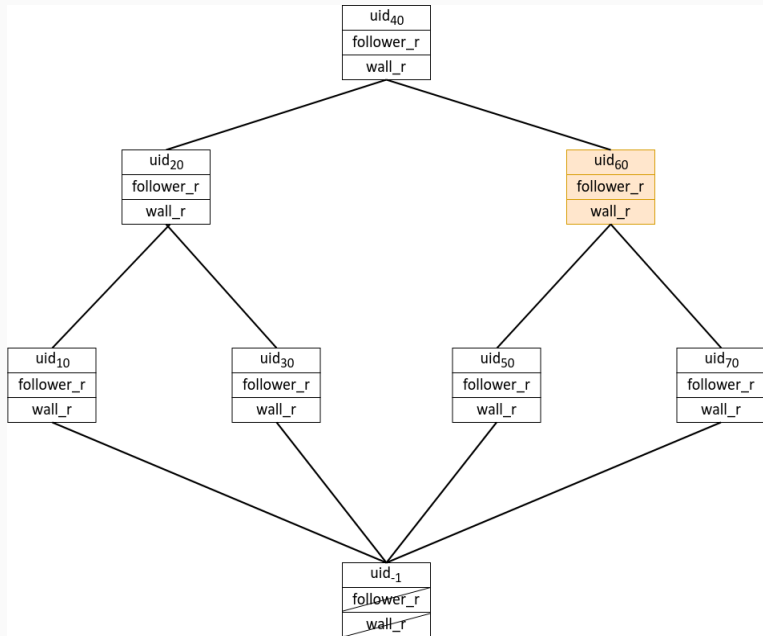
U 50 60

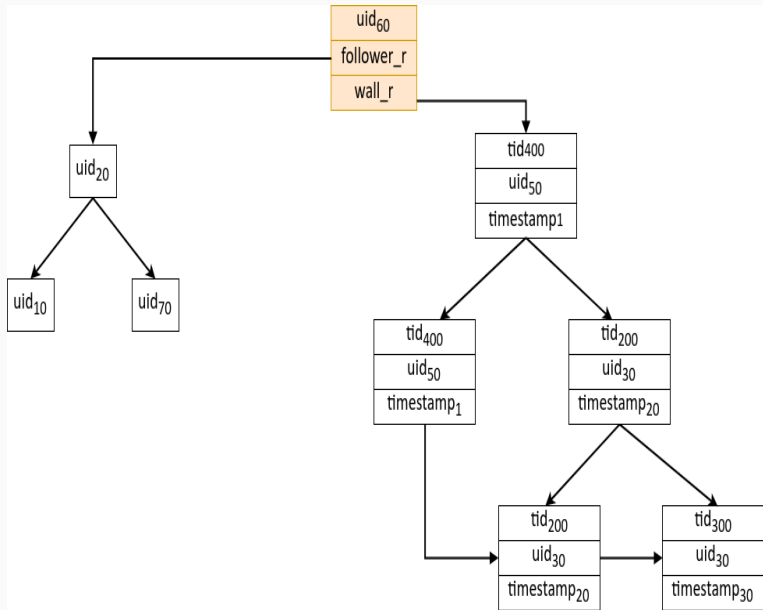


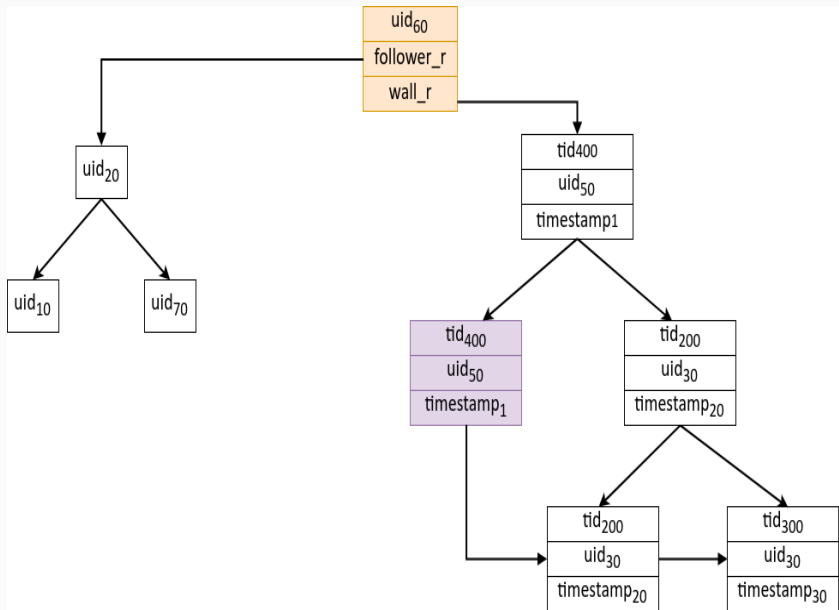


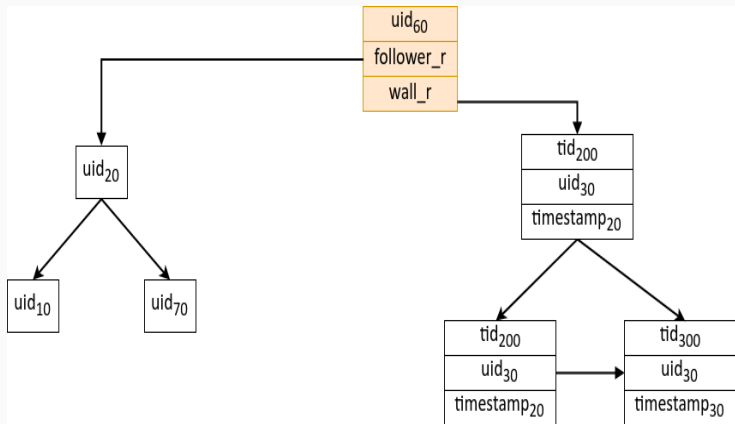












Διαγραφή του χρήστη με αναγνωριστικό *uid* από το σύστημα

Delete

D uid

Ορίσματα:

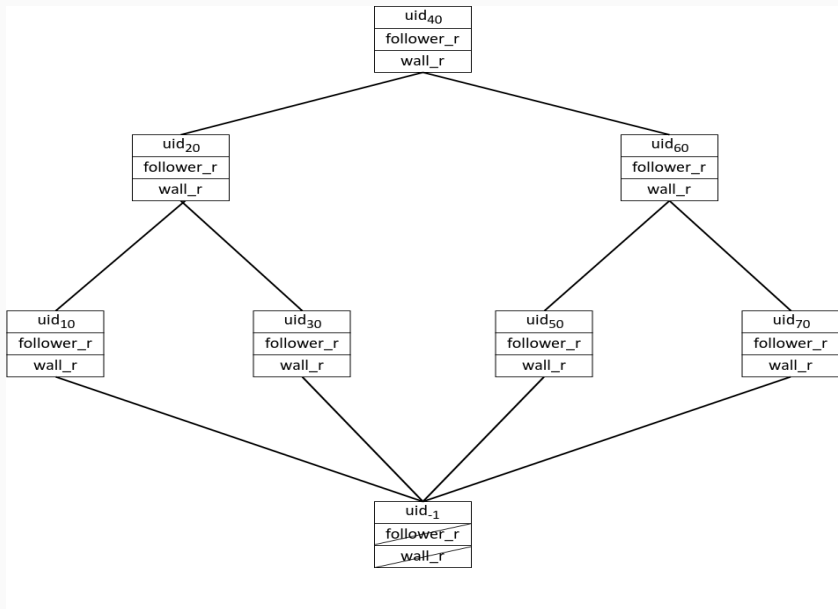
uid Αναγνωριστικό του χρήστη προς δια-
γραφή

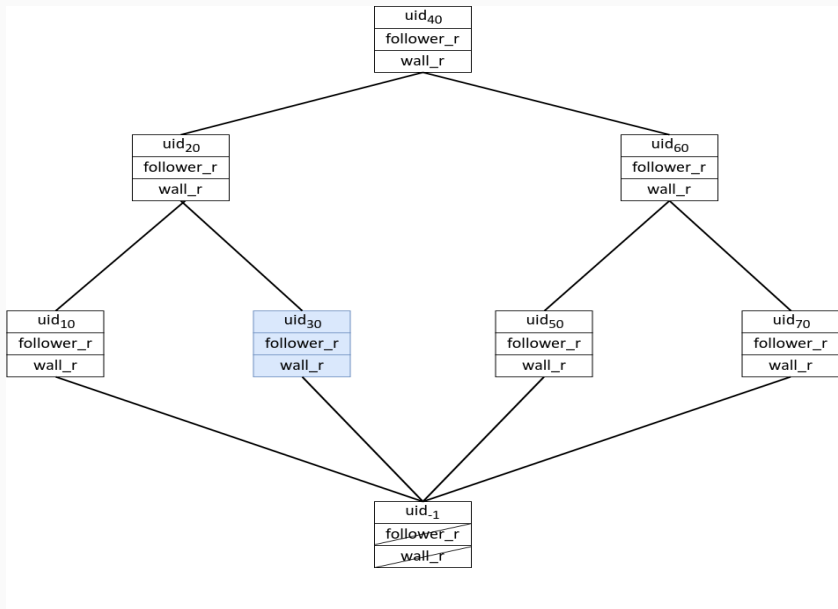
Κατά το γεγονός αυτό:

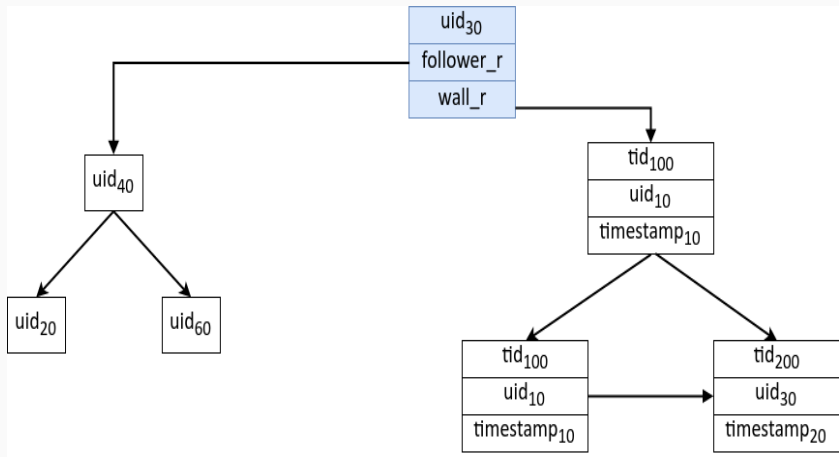
1. Αναζητούμε στο δένδρο χρηστών το χρήστη με αναγνωριστικό *uid*
2. Διαγράφουμε όλα τα στοιχεία του δένδρου των tweets αυτού του χρήστη
3. Διατρέχουμε το δένδρο των οπαδών και για κάθε κόμβο:
 - 3.1 Καλούμε κατάλληλα τη συνάρτηση που υλοποιεί το γεγονός unsubscribe
4. Διαγράφουμε από το δένδρο χρηστών το χρήστη με αναγνωριστικό *uid*
5. Τέλος διαγράφουμε όλα τα tweets του χρήστη με αναγνωριστικό *uid* από τον πίνακα κατακερματισμού των tweets θέτοντας το πεδίο *remove = true*

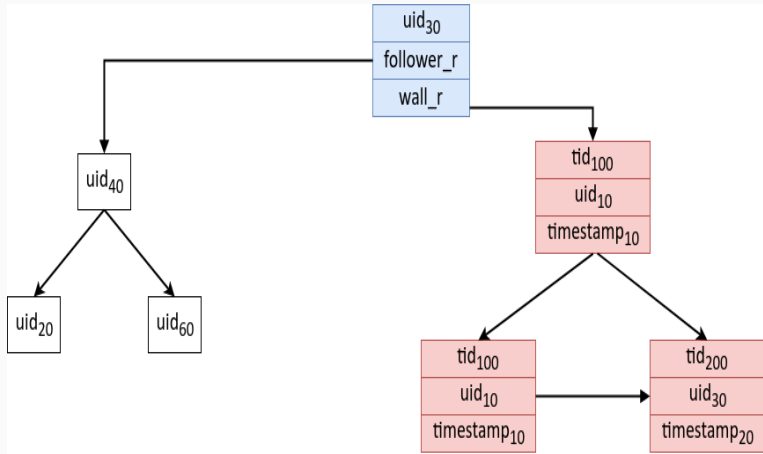
Delete

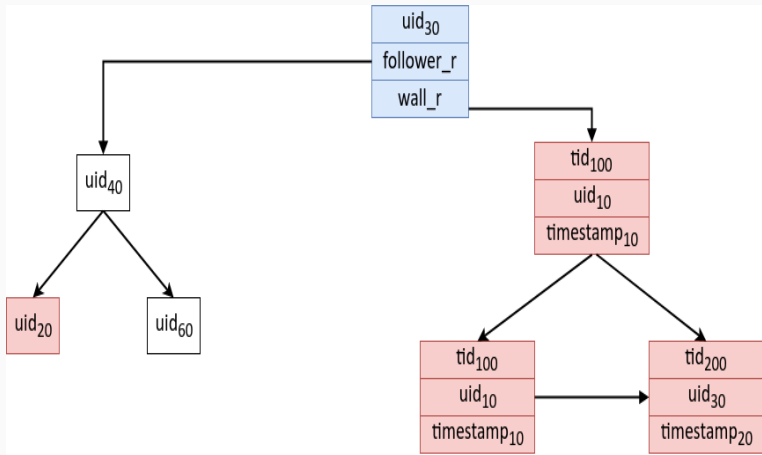
D 30

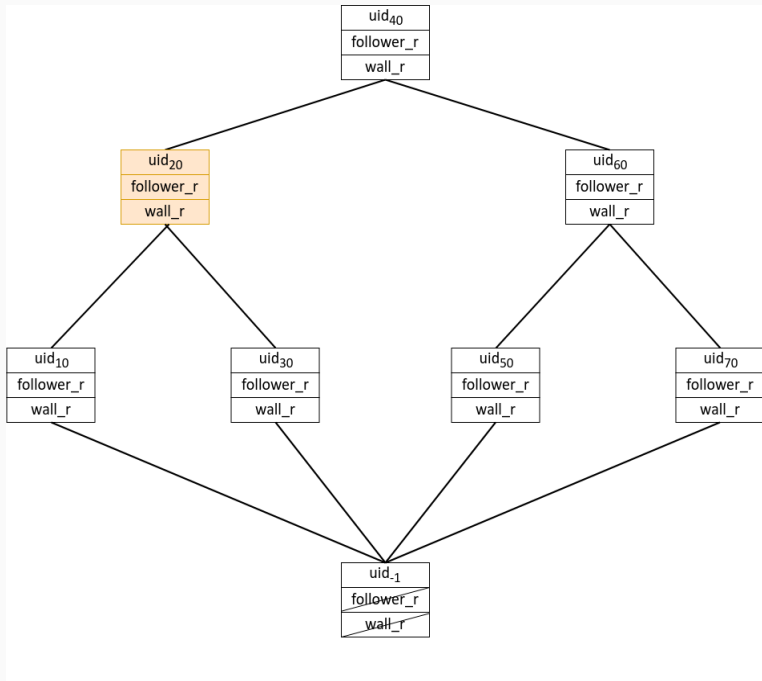


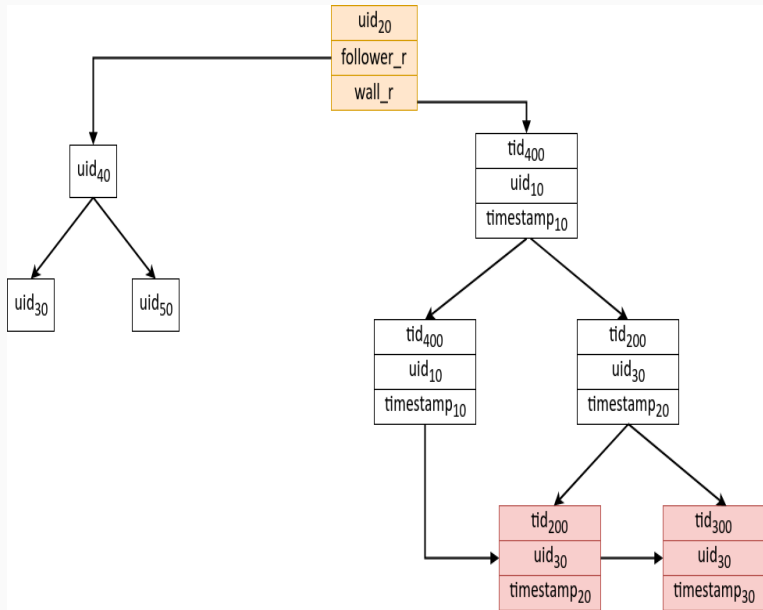


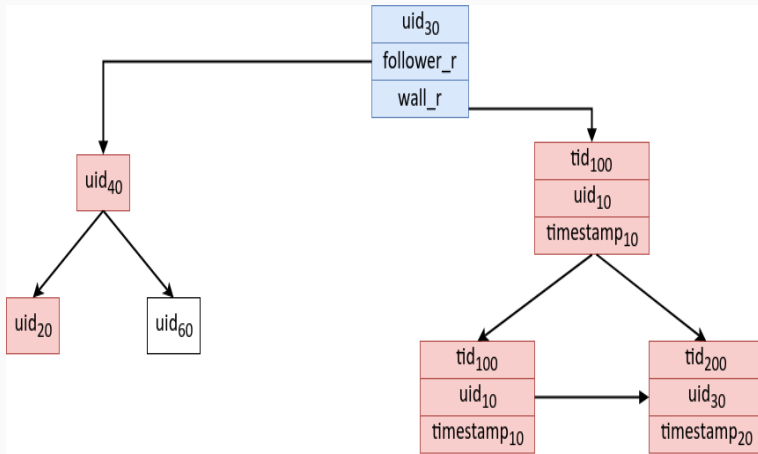


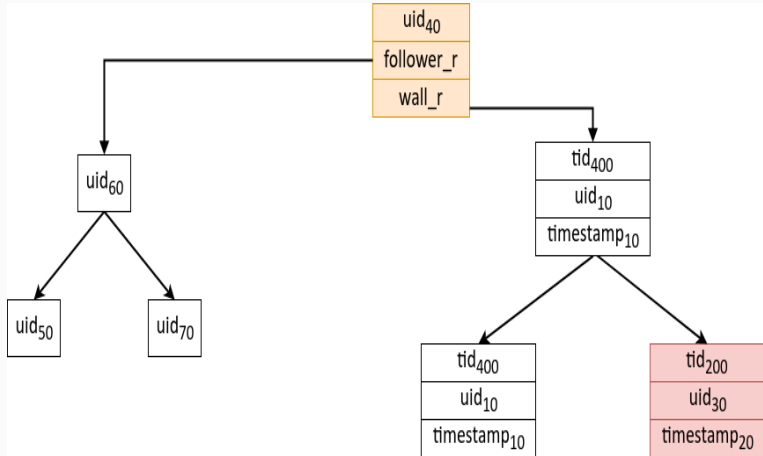


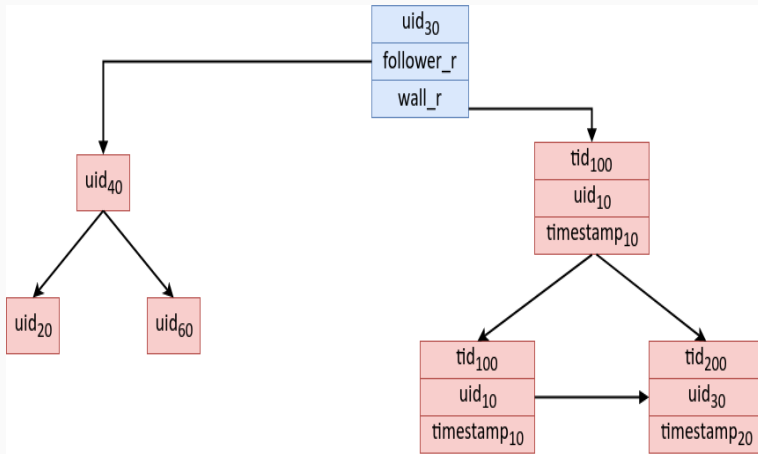


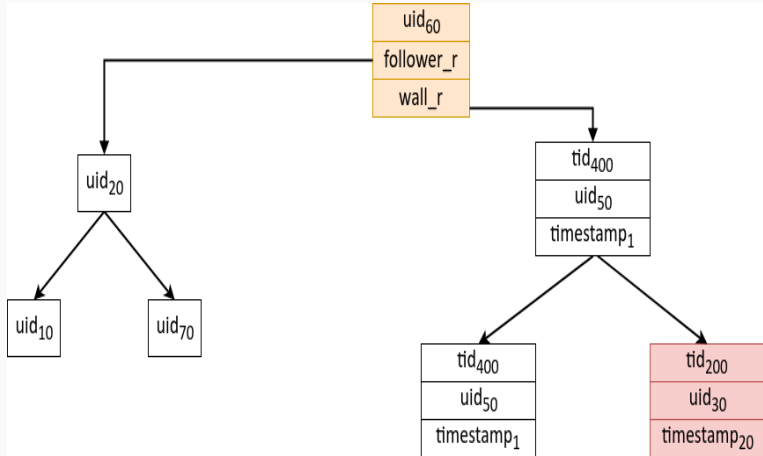












Αναζήτηση των tweets που έχει δημοσιεύσει ο χρήστης με αναγνωριστικό *uid* μεταξύ των ημερομηνιών *date₁* *date₂*

History

H uid date₁ date₂

Ορίσματα:

- uid* Αναγνωριστικό του χρήστη
- date₁* Αρχική ημερομηνία
- date₂* Τελική ημερομηνία

Κατά το γεγονός αυτό:

- Αναζητούμε στο δένδρο χρηστών το χρήστη με αναγνωριστικό *uid*
- Αναζητούμε στο δένδρο των tweets του χρήστη το tweet που έχει ημερομηνία ίση ή ακριβώς μεγαλύτερη από *date₁*
- Διατρέχουμε και εκτυπώνουμε όλα τα tweets που είναι μεταξύ του διαστήματος *date₁* και *date₂* μέσω των next δεικτών

Αναζήτηση ενός tweet με αναγνωριστικό *tid* στον πίνακα κατακερματισμού των tweets

Look Up

L tid

Ορίσματα:

tid Αναγνωριστικό του tweet

Εκτύπωση όλων των χρηστών από το δέντρο χρηστών

Print all Users

X

Εκτύπωση όλων των tweets από τον πίνακα κατακερματισμού
Tweets

Print all Tweets

Υ

Thanks!