

ΠΜΣ “Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού

και Τεχνητής Νοημοσύνης” **Ευφυής Αλληλεπίδραση με Κοινωνικά Δίκτυα**

Καθηγητής: Διονύσιος Σωτηρόπουλος

Εργασία Μαθήματος

Πέτρος Βαλαχέας - ΜΠΣΠ18003

**Β’ εξάμηνο**

**Πειραιάς 2019**

### 1. Εκφώνηση

Σκοπός της συγκεκριμένης εργασίας είναι η προγραμματιστική διαχείριση του χρονικά μεταβαλλόμενου δικτύου **Stack Overflow Temporal Network**, η περιγραφή του οποίου αλλά και τα δεδομένα του βρίσκονται στην δικτυακή τοποθεσία [https://snap.stanford.edu/data/sx-](https://snap.stanford.edu/data/sx-stackoverflow.html) [stackoverflow.html](https://snap.stanford.edu/data/sx-stackoverflow.html) . Κάθε ακμή του εν λόγω δικτύου είναι συσχετισμένη με μία χρονοσφραγίδα (timestamp) η οποία αντιστοιχεί στην χρονική στιγμή κατά την οποία δημιουργήθηκε. Το σύνολο των κατευθυνόμενων ακμών του δικτύου με τις αντίστοιχες χρονοσφραγίδες είναι αποθηκευμένο στο αρχείο **[sx-stackoverflow.txt](https://snap.stanford.edu/data/sx-stackoverflow.txt.gz)** υπό την μορφή διαδοχικών τριάδων **(source\_id, target\_id,timestamp)**, όπου το **source\_id** είναι το αναγνωριστικό του κόμβου προέλευσης της ακμής, το **target\_id** είναι το αναγνωριστικό του κόμβου κατάληξης της ακμής ενώ το **timestamp** υποδηλώνει την χρονική στιγμή δημιουργίας της ακμής.

Συνεπώς, τα διαθέσιμα δεδομένα για το υπό εξέταση δίκτυο μπορούν να αποτυπωθούν ως χρονικά συσχετισμένες ακμές της μορφής:

𝑒𝑖𝑗(𝑡) = 〈𝑣𝑖, 𝑣𝑗, 𝑡〉 (1) για 𝑡𝑚𝑖𝑛 ≤ 𝑡 ≤ 𝑡𝑚𝑎𝑥

όπου 𝑡𝑚𝑖𝑛 είναι η παλιότερη χρονική παρατήρηση που υπάρχει μέσα στο σύνολο των διαθέσιμων δεδομένων και 𝑡𝑚𝑎𝑥 η πιο πρόσφατη χρονική παρατήρηση. Το συγκεκριμένο χρονικό διάστημα 𝑇 = [𝑡𝑚𝑖𝑛, 𝑡𝑚𝑎𝑥] διαμερίζεται σε ένα σύνολο (𝑁) μη-επικαλυπτόμενων χρονικών περιόδων {𝑇1, 𝑇2, ⋯ , 𝑇𝑗, … , 𝑇𝑁} ίσης χρονικής διάρκειας (𝛿𝑡) θεωρώντας ένα σύνολο

(𝑁 + 1) χρονικών στιγμών {𝑡0, 𝑡1, 𝑡2, ⋯ , 𝑡𝑗−1, 𝑡𝑗 … , 𝑡𝑁−1, 𝑡𝑁} τέτοιων ώστε:

𝑡𝑗 = 𝑡𝑚𝑖𝑛 + 𝑗 ∗ 𝛿𝑡 (2) για 0 ≤ 𝑗 ≤ 𝑁

όπου 𝛥𝑇 = 𝑡𝑚𝑎𝑥 − 𝑡𝑚𝑖𝑛

(3) και 𝛿𝑡 = 𝛥𝑇 (4). Σύμφωνα με τις παραπάνω διευκρινήσεις η j-

οστή χρονική περίοδος μπορεί να οριστεί σύμφωνα με την παρακάτω σχέση:

[𝑡𝑗−1, 𝑡𝑗), 1 ≤ 𝑗 ≤ 𝑁 − 1;

𝑇𝑗 = {

[𝑡𝑗−1, 𝑡𝑗], 𝑗 = 𝑁.

(5)

Για κάθε μία από τις χρονικές περιόδους 𝑇𝑗 για 1 ≤ 𝑗 ≤ 𝑁 μπορούμε να θεωρήσουμε το αντίστοιχο υπο-γράφημα του συνολικού δικτύου 𝐺[𝑡𝑗−1, 𝑡𝑗] = (𝑉[𝑡𝑗−1, 𝑡𝑗], 𝐸[𝑡𝑗−1, 𝑡𝑗]) όπου

𝑉[𝑡𝑗−1, 𝑡𝑗] είναι το σύνολο των κορυφών που εμφανίζονται στα άκρα των ακμών του δικτύου

κατά την χρονική περίοδο 𝑇𝑗. Το σύνολο των ακμών του δικτύου που δημιουργούνται την συγκεκριμένη χρονική περίοδο είναι το σύνολο 𝐸[𝑡𝑗−1, 𝑡𝑗].

Η αυστηρότερη περιγραφή της χρονικής εξέλιξης του εξεταζόμενου δικτύου μέσα στο πλαίσιο του προβλήματος της πρόγνωσης ακμών επιβάλλει την διατύπωση μερικών συμπληρωματικών σχέσεων. Συγκεκριμένα, κατά την μετάβαση του δικτύου από την χρονική περίοδο 𝑇𝑗 στην χρονική περίοδο 𝑇𝑗+1 μας ενδιαφέρει το σύνολο των κορυφών που παραμένει κοινό μεταξύ

των χρονικών διαστημάτων [𝑡𝑗−1, 𝑡𝑗] και [𝑡𝑗, 𝑡𝑗+1], το οποίο θα υποδηλώνεται ώς το σύνολο

𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] που θα δίνεται από την σχέση:

𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] = 𝑉 [𝑡𝑗−1, 𝑡𝑗]⋂𝑉[𝑡𝑗, 𝑡𝑗+1](6) για 1 ≤ 𝑗 ≤ 𝑁 − 1.

Αντίστοιχα, μας ενδιαφέρει ο περιορισμός των συνόλων 𝐸[𝑡𝑗−1, 𝑡𝑗] και 𝐸[𝑡𝑗, 𝑡𝑗+1] σε εκείνα τα υποσύνολα των ακμών που οι κορυφές τους ανήκουν αυστηρά στο σύνολο 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1]. Αυτά τα περιορισμένα σύνολα ακμών θα υποδηλώνονται ως το σύνολο 𝐸∗[𝑡𝑗, 𝑡𝑗+1] και θα δίνονται από τις σχέσεις:

𝐸∗[𝑡𝑗−1, 𝑡𝑗] = {(𝑢, 𝑣) ∈ 𝐸[𝑡𝑗−1, 𝑡𝑗]: 𝑢 ∈ 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] 𝜅𝛼𝜄 𝑣 ∈ 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1]} (7)

𝐸∗[𝑡𝑗, 𝑡𝑗+1] = {(𝑢, 𝑣) ∈ 𝐸[𝑡𝑗, 𝑡𝑗+1]: 𝑢 ∈ 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] 𝜅𝛼𝜄 𝑣 ∈ 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1]} (8)

Η προγραμματιστική διαχείριση του προαναφερθέντος χρονικά μεταβαλλόμενου δικτύου συνίστασται στην συγγραφή κώδικα είτε στο προγραμματιστικό περιβάλλον του **MatLab** είτε της **Python** προκειμένου να υλοποιηθούν οι ακόλουθες διαδικασίες:

1. Υπολογισμός των χρονικών στιγμών 𝑡𝑚𝑖𝑛 και 𝑡𝑚𝑎𝑥.
2. Διαμέριση του συνολικού χρονικού διαστήματος 𝑇 = [𝑡𝑚𝑖𝑛, 𝑡𝑚𝑎𝑥] στα υποδιαστήματα

{𝑇1, 𝑇2, ⋯ , 𝑇𝑗, … , 𝑇𝑁} και υπολογισμός των αντίστοιχων χρονικών στιγμών

{𝑡0, 𝑡1, 𝑡2, ⋯ , 𝑡𝑗−1, 𝑡𝑗 … , 𝑡𝑁−1, 𝑡𝑁} συναρτήσει της παραμέτρου (𝑁). Η παράμετρος (𝑁) θα μπορεί να μεταβληθεί από τον χρήστη του προγράμματος πριν από την εκτέλεσή του.

1. Προγραμματιστική αποτύπωση (είτε μέσω της μήτρας γειτνίασης είτε μέσω κάποιου εγγενούς για

το εργαλείο που θα χρησιμοποιήσετε τρόπου, π.χ. ενός αντικειμένου Graph του module NetworkX της Python) του συνόλου των υποδικτύων 𝐺[𝑡𝑗−1, 𝑡𝑗] για 1 ≤ 𝑗 ≤ 𝑁.

1. Για κάθε ένα από τα υποδίκτυα 𝐺[𝑡𝑗−1, 𝑡𝑗] για 1 ≤ 𝑗 ≤ 𝑁 να υπολογίσετε και να παρουσιάσετε γραφικά την κατανομή των τιμών των παρακάτω μέτρων κεντρικότητας:
   1. Degree Centrality
   2. In-Degree Centrality
   3. Out-Degree Centrality
   4. Closeness Centrality
   5. Betweenness Centrality
   6. Eigenvector Centrality
   7. Katz Centrality

5. Για κάθε ζεύγος διαδοχικών υποδικτύων (𝐺[𝑡𝑗−1, 𝑡𝑗], 𝐺[𝑡𝑗, 𝑡𝑗+1]) για 1 ≤ 𝑗 ≤ 𝑁 − 1να υπολογιστούν τα σύνολα 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1], 𝐸∗[𝑡𝑗−1, 𝑡𝑗] και 𝐸∗[𝑡𝑗, 𝑡𝑗+1].

6. Για κάθε ζεύγος κόμβων (𝑢, 𝑣) ∈ 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] και κάθε σύνολο 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] με 1 ≤ 𝑗 ≤ 𝑁 − 1να υπολογιστούν οι παρακάτω πίνακες ομοιότητας:

i. 𝑆𝐺𝐷 = [𝑆𝐺𝐷(𝑢, 𝑣)] = −𝐿𝑒𝑛𝑔𝑡ℎ 𝑜𝑓 𝑆ℎ𝑜𝑟𝑡𝑒𝑠𝑡 𝑃𝑎𝑡ℎ 𝐵𝑒𝑡𝑤𝑒𝑒𝑛 𝑢 𝑎𝑛𝑑 𝑣

[**Graph Distance**]

ii. 𝑆𝐶𝑁 = [𝑆𝐶𝑁 (𝑢, 𝑣)] = |𝛤(𝑢)⋂𝛤(𝑣)| [**Common Neighbors**] όπου 𝛤(𝑢) το σύνολο των γειτόνων του κόμβου 𝑢.

1. Jaccard’sCoefficient
2. Adamic / Adar
3. Preferential Attachment

### Προσοχή!!! Οι παραπάνω πίνακες ομοιότητας θα υπολογίζονται για το σύνολο των κόμβων που είναι κοινοί για δύο διαδοχικά υποδίκτυα, δηλαδή το σύνολο 𝑽∗[𝒕𝒋−𝟏, 𝒕𝒋+𝟏], αλλά πάνω στην βάση του συνόλου των ακμών 𝑬∗[𝒕𝒋−𝟏, 𝒕𝒋]. Πρόκειται για τις ακμές της προηγούμενης χρονικής περιόδου που σχηματίζονται όμως μεταξύ κορυφών που ανήκουν στο κοινό σύνολο των κόμβων 𝑽∗[𝒕𝒋−𝟏, 𝒕𝒋+𝟏].

7. Για κάθε έναν από τους πίνακες ομοιότητας SGD, SCN, SJC, SAκαι SPA που υπολογίστηκαν στο προηγούμενο ερώτημα, για κάθε ένα από τα σύνολα κόμβων 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1], να εξαχθούν οι

κορυφαίες 𝑝𝐺𝐷%, 𝑝𝐶𝑁%,𝑝𝐽𝐶 %,𝑝𝐴% και 𝑝𝑃𝐴% (μεγαλύτερες) τιμές ομοιότητας και τα ζεύγη των κορυφών στις οποίες αντιστοιχούν. Το ποσοστό αυτών των ζευγαριών των κορυφών που ανήκουν πράγματι στο σύνολο 𝐸∗[𝑡𝑗, 𝑡𝑗+1] υποδηλώνει το ποσοστό επιτυχίας στην πρόγνωση μελλοντικών ακμών της κάθε μετρικής. Να υπολογιστούν τα ποσοστό ορθής πρόγνωσης για κάθε μέτρο ομοιότητας για κάθε σύνολο 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] (δηλαδή για κάθε ζεύγος διαδοχικών υποδικτύων). Οι

τιμές των παραμέτρων 𝑝𝐺𝐷%, 𝑝𝐶𝑁%,𝑝𝐽𝐶 %,𝑝𝐴% και 𝑝𝑃𝐴% θα δίνονται από τον χρήστη του προγράμματος πριν από την εκτέλεση του προγράμματος.

### Περιγραφή προγράμματος

Για την εκπόνηση του προγράμματος χρησιμοποιήθηκαν οι βιβλιοθήκες NetworkX1, matplotlib2 και NumPy3.

Επίσης κατά το διάβασμα του αρχείου χρησιμοποιείται η τιμή μιας μεταβλητής (N\_RL) που καθορίζει το πόσες γραμμές θα διαβαστούν καθώς το διάβασμα ολόκληρου του αρχείου αποδείχθηκε πολύ χρονοβόρο.

* 1. *Κύρια ροή*

Αρχικά καλείται η συνάρτηση min\_max η οποία υπολογίζει το μέγιστο και το ελάχιστο timestamp. Στη συνέχεια το πρόγραμμα χωρίζει το συνολικό χρονικό διάστημα σε υποδιαστήματα συναρτήσει της παραμέτρου *Ν* που ορίζεται στην αρχή του προγράμματος. Τέλος για κάθε υποδιάστημα σχεδιάζεται το γράφημα, το ιστόγραμμα της κεντρικότητας που επιλέγεται, υπολογίζονται τα σύνολα του 5ου ερωτήματος, υπολογίζονται οι πίνακες ομοιότητας και επίσης τα ποσοστά ορθής πρόγνωσης του κάθε πίνακα ομοιότητας.

* 1. *Συνάρτηση min\_max*

Σε κάθε περίπτωση η συνάρτηση διαβάζει ένα αρχείο με όνομα “min\_maxN.txt” όπου N η παράμετρος του προγράμματος που ορίζει πόσες γραμμές διαβάζονται (π.χ. min\_max10000.txt). Αν βρεθεί το αρχείο τότε οι τιμές διαβάζονται από αυτό εναλλακτικά βρίσκει τις τιμές εκ νέου. Για να βρει τις τιμές αρχικά ανοίγει το αρχείο και διαβάζει ανά γραμμή. Κάθε timestamp συγκρίνεται με την μέγιστη και την ελάχιστη τιμή και αυτές ανανεώνονται. Όταν τελειώσει η αναζήτηση τυπώνει το αποτέλεσμα σε αρχείο και επιστρέφει το μέγιστο και το ελάχιστο.

1 networkx.github.io

2 matplotlib.org

3 [www.numpy.org](http://www.numpy.org/)

5

* 1. *Συνάρτηση graph*

Αυτή η συνάρτηση καλείται από την κύρια ροή του προγράμματος και είναι υπεύθυνη για την δημιουργία και τον σχεδιασμό ενός γραφήματος. Παίρνει ως όρισμα δύο μεταβλητές: την *t* που ορίζει το υποσύνολο του χρόνου για το οποίο δημιουργείται το γράφημα και την *centrality* που ορίζει ποιο μέτρο κεντρικότητας θα χρησιμοποιηθεί για την δημιουργία του ιστογράμματος βαθμών στη συνέχεια.

Δημιουργεί αρχικά ένα κενό κατευθυνόμενο γράφημα και διαβάζει το αρχείο ανά γραμμή. Αν το timestamp ανήκει στο υποσύνολο του χρόνου για το οποίο έχει κληθεί η συνάρτηση και αν το source\_id και το target\_id είναι διαφορετικά (δηλαδή η γραμμή δεν περιγράφει ένα self- loop) τότε προστίθεται μία ακμή στο γράφημα από τον κόμβο source\_id στο κόμβο target\_id. Όταν τελειώσει το διάβασμα τότε γίνεται γραφική αναπαράσταση του γραφήματος και η συνάρτηση καλεί την συνάρτηση centrality\_histogram.

* 1. *Συνάρτηση centrality\_histogram*

Η συνάρτηση καλείται από την συνάρτηση graph και είναι υπεύθυνη για τον υπολογισμό και την γραφική απεικόνιση σε ιστόγραμμα των μέτρων κεντρικότητας των γραφημάτων. Παίρνει ως όρισμα δύο μεταβλητές: την *g* που είναι ένα αντικείμενο γραφήματος της βιβλιοθήκης NetworkX και την *c* που ορίζει το μέτρο κεντρικότητας που θα χρησιμοποιηθεί. Η τελευταία μπορεί να πάρει τις τιμές ‘degree’, ‘in\_degree’, ‘out\_degree’, closeness’, ‘betweenness’, ‘eigenvector’, ‘katz’.

Βρίσκει τους βαθμούς και των αριθμό εμφανίσεων κάθε βαθμού και στη συνέχεια τους απεικονίζει γραφικά σε ένα ιστόγραμμα.

* 1. *Συνάρτηση graph\_star*

Η συνάρτηση καλείται από την κύρια ροή του προγράμματος και υπολογίζει τα σύνολα

𝑉∗[𝑡𝑗−1, 𝑡𝑗+1], 𝐸∗[𝑡𝑗−1, 𝑡𝑗] και 𝐸∗[𝑡𝑗, 𝑡𝑗+1]. Παίρνει ως όρισμα την μεταβλητή *t* που ορίζει το υποσύνολο του χρόνου για το οποίο καλείται η συνάρτηση.

Η συνάρτηση αρχικά δημιουργεί δύο κατευθυνόμενα γραφήματα που αντιστοιχούν στα

υποδίκτυα (𝐺[𝑡𝑗−1, 𝑡𝑗], 𝐺[𝑡𝑗, 𝑡𝑗+1]). Στη συνέχεια συγκρίνεται κάθε κόμβος του πρώτου γραφήματος με του δεύτερου και στην περίπτωση που είναι ίδιοι τότε προστίθεται στο σύνολο V\*. Επίσης για κάθε ακμή σε καθένα από τα υποδίκτυα (𝐺[𝑡𝑗−1, 𝑡𝑗], 𝐺[𝑡𝑗, 𝑡𝑗+1]) ελέγχεται αν και οι δύο της κόμβοι βρίσκονται στο σύνολο V\*. Σε αυτή την περίπτωση η ακμή προστίθεται στο σύνολο 𝐸∗[𝑡𝑗−1, 𝑡𝑗] ή 𝐸∗[𝑡𝑗, 𝑡𝑗+1] ανάλογα με το από πού προήλθε η ακμή και τέλος τυπώνονται νέα τρία σύνολα και καλείται η συνάρτηση similarity\_matrices.

* 1. *Συνάρτηση similarity\_matrices*

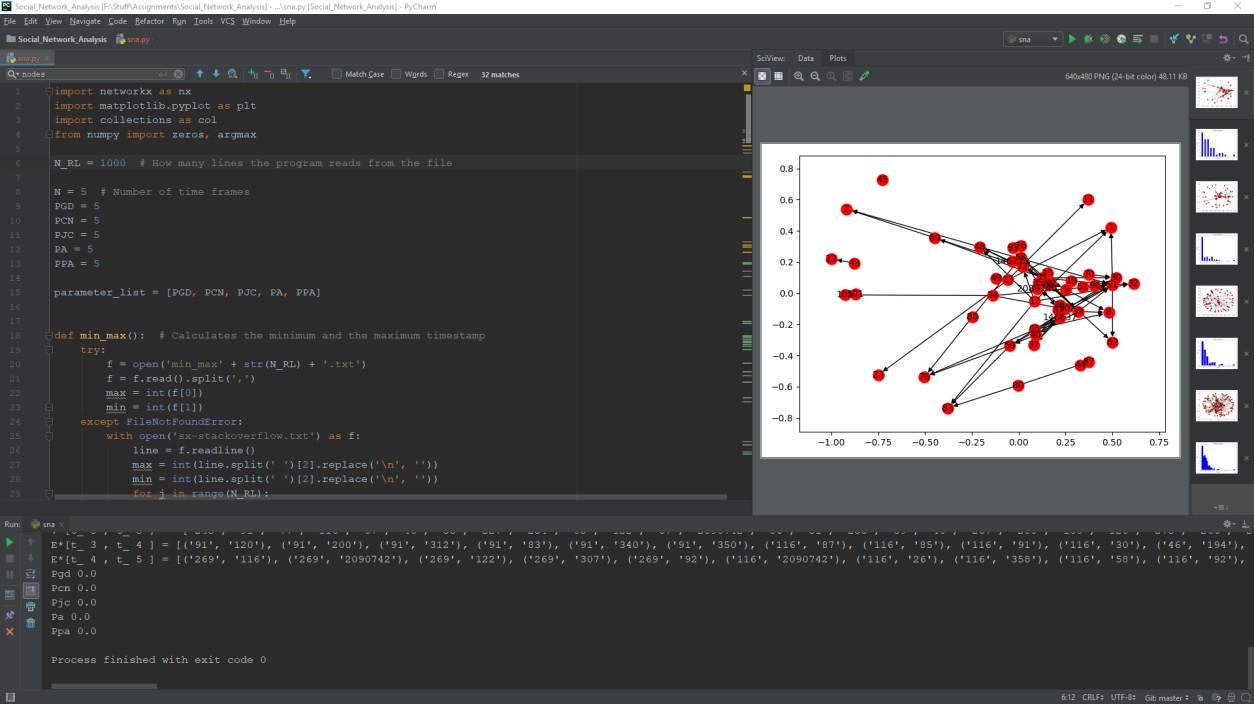
Η συνάρτηση καλείται από την συνάρτηση graph\_star, υπολογίζει τους πίνακες ομοιότητας, τα ποσοστά επιτυχίας πρόβλεψης και παίρνει ως όρισμα μία μεταβλητή *edges* η οποία αντιστοιχεί στο σύνολο των ακμών 𝐸∗[𝑡𝑗−1, 𝑡𝑗] που υπολογίζεται από την συνάρτηση graph\_star και μία μεταβλητή *nodes* η οποία αντιστοιχεί στο σύνολο κόμβων 𝑉∗[𝑡𝑗−1, 𝑡𝑗+1].

Αρχικά ανακατασκευάζει ένα κατευθυνόμενο γράφημα με ακμές αυτές του ορίσματος και επίσης δημιουργεί το αντίστοιχο μη-κατευθυνόμενο γράφημα που χρειάζεται για τον υπολογισμό των περισσότερων πινάκων ομοιότητας. Για κάθε ζεύγος κόμβων (𝑢, 𝑣) ∈

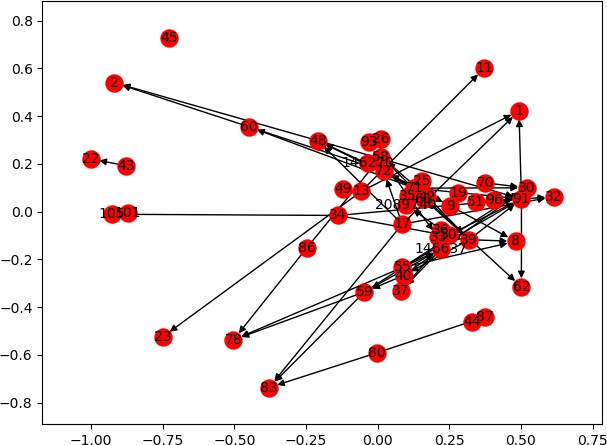
𝑉∗[𝑡𝑗−1, 𝑡𝑗+1] υπολογίζει την τιμή της εκάστοτε μετρικής ομοιότητας και την εκχωρεί σε κατάλληλο πίνακα. Τέλος βρίσκει τις n μεγαλύτερες τιμές των πινάκων, (όπου n μία παράμετρος που ορίζεται στην αρχή του προγράμματος και είναι διαφορετική για κάθε μετρική ομοιότητας) σε ποιους κόμβους αντιστοιχούν και υπολογίζει τον λόγο: αριθμός ακμών που αντιστοιχούν στις μεγαλύτερες τιμές και ανήκουν στο σύνολο 𝐸∗[𝑡𝑗−1, 𝑡𝑗] προς συνολικός αριθμός ακμών που αντιστοιχούν στις μεγαλύτερες τιμές.

### Παραδείγματα εκτέλεσης

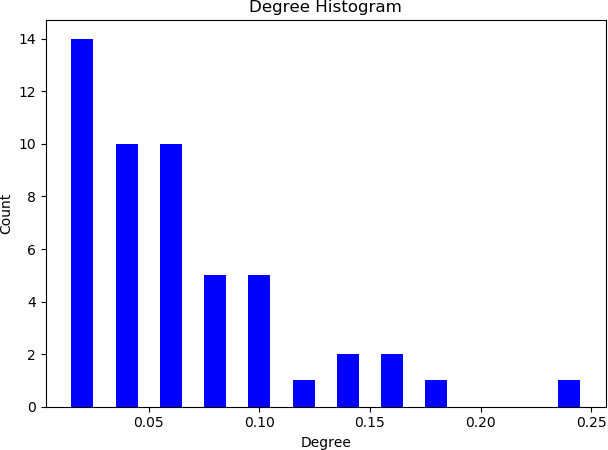
* 1. Τιμές παραμέτρων



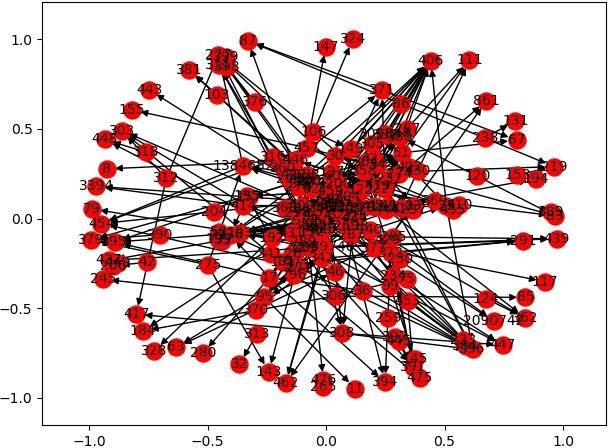
* 1. Συνολική εικόνα παραθύρου



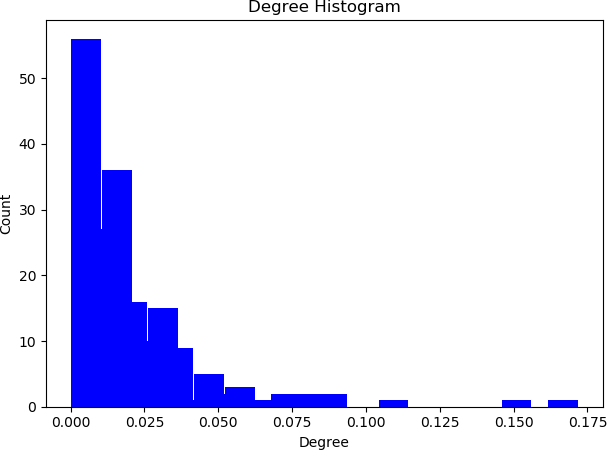
* 1. Γράφημα για t = 0



* 1. Αντίστοιχο ιστόγραμμα για degree centrality



* 1. Γράφημα για t = 4



* 1. Αντίστοιχο ιστόγραμμα για degree centrality