

Πανεπιστήμιο Πειραιώς

Τμήμα Πληροφορικής



ΕΥΦΥΗΣ ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕ ΚΟΙΝΩΝΙΚΑ ΔΙΚΤΥΑ

Χατζηκοκολάκης Κωνσταντίνος-Νικόλαος Π15186

Ιωαννίδης Σωκράτης Π15045

2 Ιουλίου 2018

Contents

1	Εισαγωγή	3
2	Υλοποίηση	4
2.1	Ερώτημα 1	4
2.2	Ερώτημα 2	4
2.3	Ερώτημα 3	5
2.4	Ερώτημα 4	8
2.5	Ερώτημα 5	15
2.6	Ερώτημα 6	15
2.7	Ερώτημα 7	16
2.8	Παράδειγμα εκτέλεσης	16

List of Figures

1	A subgraph	5
2	A subgraph	6
3	A subgraph	7
4	Degree Centrality	8
5	InDegree Centrality	9
6	OutDegree Centrality	10
7	Closeness Centrality	11
8	Betweenness Centrality	12
9	Eigenvector Centrality	13
10	Katz Centrality	14
11	Παράδειγμα εκτέλεσης του προγράμματος	16

1 Εισαγωγή

Στην παρούσα εργασία καλούμαστε να υλοποιήσουμε ένα πρόγραμμα το οποίο βασισμένο στο δίκτυο του Stack Overflow, εξάγει κάποιες μετρικές κεντρικότητας και κάποια στατιστικά, όπως για παράδειγμα την πρόβλεψη ύπαρξης δεσμού μεταξύ δύο ακμών σε ένα γράφημα σε μια μελλοντική χρονική στιγμή.

Η υλοποίηση έγινε σε Python3.

Το δίκτυο του Stack overflow μπορεί να βρεθεί στον παρακάτω σύνδεσμο:

<https://snap.stanford.edu/data/sx-stackoverflow.html>

Το πρόγραμμα αρχικά διαβάζει από αρχείο txt του κόμβους του γραφήματος, με τις συνδέσεις μεταξύ τους να γίνονται σε συγκεκριμένες χρονικές στιγμές, και κατόπιν ζητάει από τον χρήστη να εισάγει τον αριθμό των διαστημάτων στον οποίο επιθυμεί να σπάσει το αρχικό διάστημα. Όσος είναι ο αριθμός αυτός, τόσα θα είναι και τα υποδιαστήματα που θα διαμεριστεί το αρχικό διάστημα. Κατόπιν, για κάθε υποδιάστημα, δημιουργείται ένα καινούριο υπογράφημα, στο οποίο υπολογίζονται οι πίνακες E^* , V^* , οι μετρικές ομοιότητας καθώς και οι μετρικές κεντρικότητας.

2 Υλοποίηση

2.1 Ερώτημα 1

Σε αυτό το ερώτημα καλούμαστε να υπολογίσουμε τα t_{min} και t_{max} .

Αυτό γίνεται κατευθείαν από το .txt αρχείο που εισάγεται στο πρόγραμμα, καθώς το αρχείο είναι ταξινομημένο ως προς την χρονική στιγμή που δημιουργήθηκε ο εκάστοτε δεσμός.

2.2 Ερώτημα 2

Σε αυτό το ερώτημα καλούμαστε να υλοποιήσουμε την διαμέριση του αρχικού διαστήματος σε N ισομήκη διαστήματα.

Ο χρήστης δίνει την τιμή που θέλει για το N στο πρόγραμμα (γίνεται έλεγχος τιμών ώστε το N που θα δοθεί να είναι τουλάχιστον 2).

Κατόπιν, το διάστημα $t_{max} - t_{min}$ διαμερίζεται σε N ισομήκη υποδιαστήματα μήκους:

$$\frac{t_{max} - t_{min}}{N}$$

Είναι αναπόφευκτο το πρόβλημα της κοπής διαστημάτων σε υποδιαστήματα με άκρα ακέραιους αριθμούς ώστε να μην προκύπτουν δεκαδικά, ωστόσο αυτό δεν αποτελεί πρόβλημα για την εκτέλεση του προβλήματος.

2.3 Ερώτημα 3

Σε αυτό το ερώτημα καλούμαστε να αναπαραστήσουμε κάθε υπογράφημα του G που παράγεται. Αυτό γίνεται με έτοιμη συνάρτηση της βιβλιοθήκης `networkx`. Αρχικά, μέσα σε ένα `for loop` υπολογίζονται ποιες γραμμές του `.txt` αρχείου αντιστοιχούν στο εκάστοτε υπογράφημα.

Για κάθε υπογράφημα, οι κόμβοι και οι δεσμοί του τοποθετούνται σε έναν πίνακα. Βάσει αυτών των πινάκων θα υπολογιστούν αργότερα τα E^* , V^* , καθώς επίσης θα μπορούν να ανακτηθούν όλα τα υπογραφήματα ανά πάσα στιγμή, καθώς τα υπογραφήματα δεν αποθηκεύονται στην μνήμη του υπολογιστή, αλλά υπολογίζουμε τη στιγμή της κατασκευής τους ότι χρειάζεται.

Παρακάτω ακολουθούν μερικά τυχαία παραδείγματα γραφημάτων που προέκυψαν από εκτελέσεις του προγράμματος.

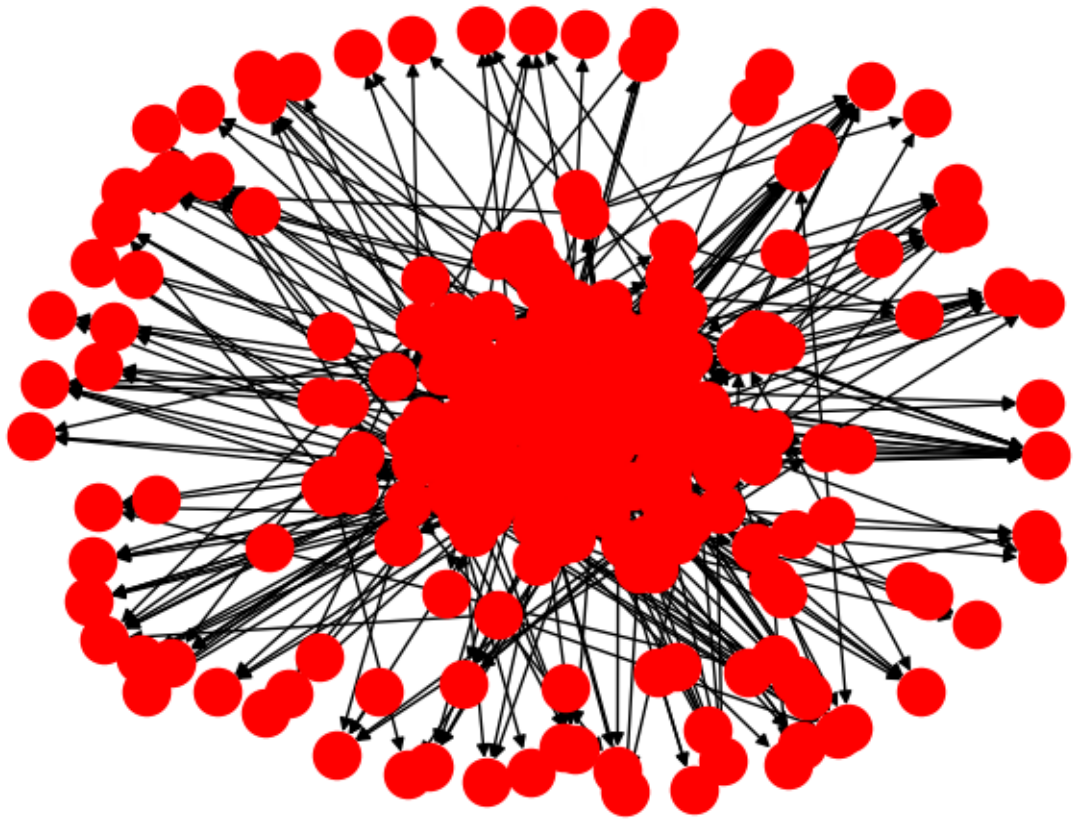


Figure 1: A subgraph

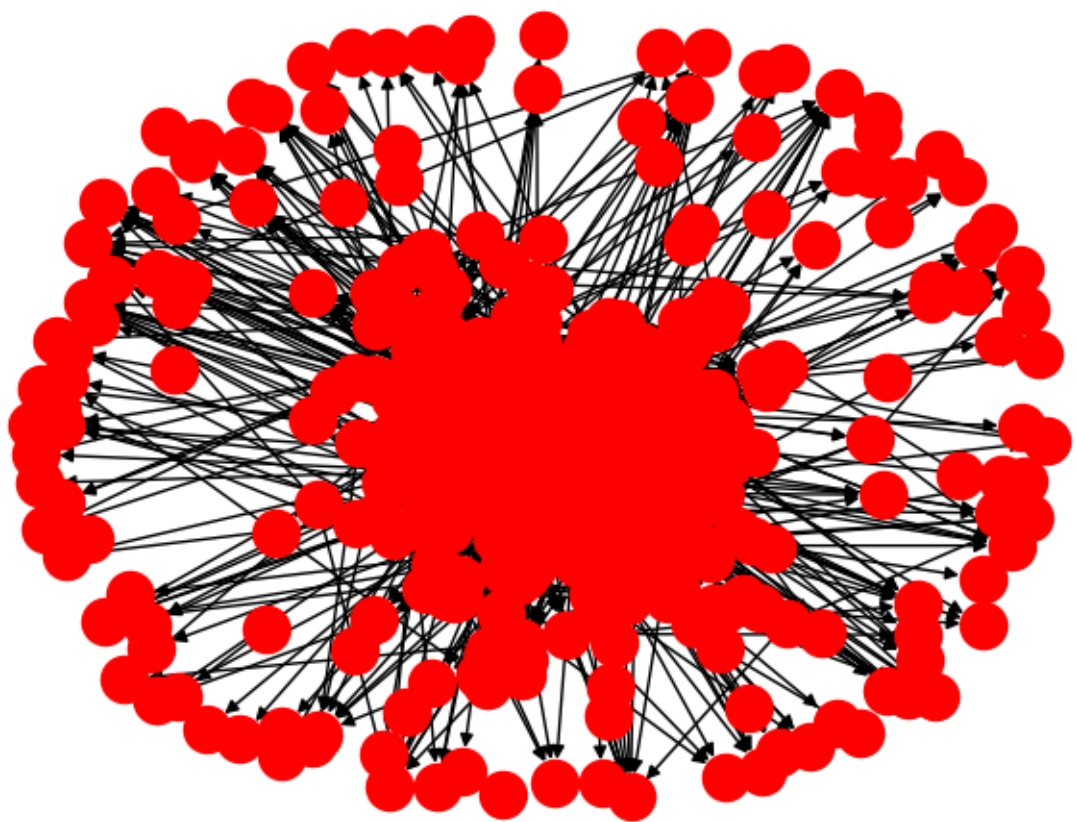


Figure 2: A subgraph

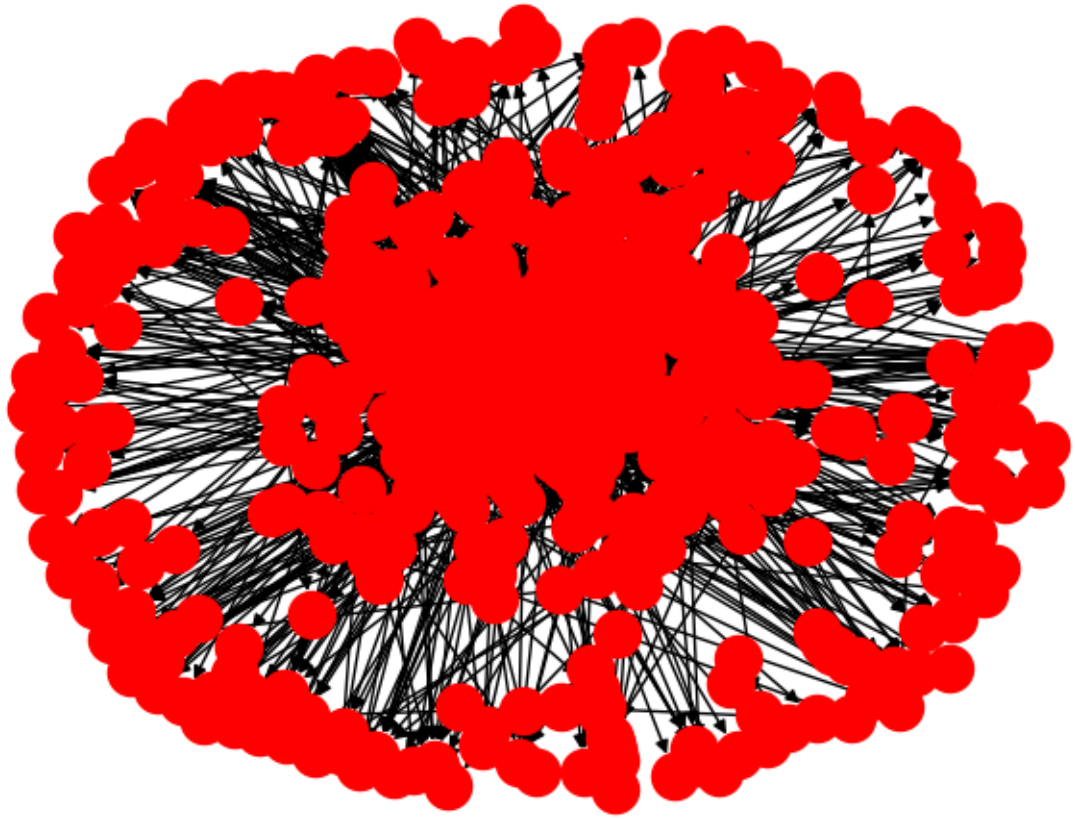


Figure 3: A subgraph

2.4 Ερώτημα 4

Στο ερώτημα αυτό ζητείται από εμάς να βρούμε τις βασικές μετρικές κεντρικότητας για κάθε υπογράφημα.

Τα μέτρα αυτά βρίσκονται κατευθείαν από έτοιμες συναρτήσεις της βιβλιοθήκης `networkx`. Απαραίτητη προϋπόθεση είναι το γράφημα να είναι ακατεύθυντο, οπότε το μετατρέπουμε σε ακατεύθυντο πρώτου βρεθούν οι μετρικές κεντρικότητας.

Παρακάτω δίνονται ενδεικτικά κάποιες εικόνες μετρικών κεντρικότητας από ένα τυχαίο παράδειγμα εκτέλεσης.

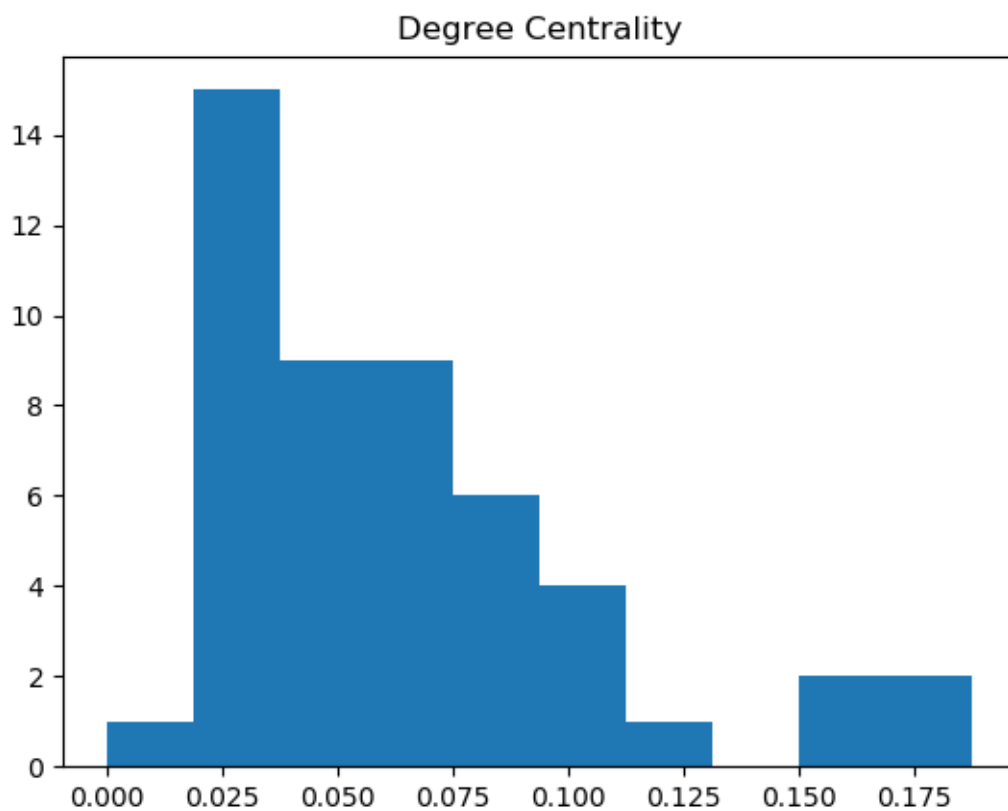


Figure 4: Degree Centrality

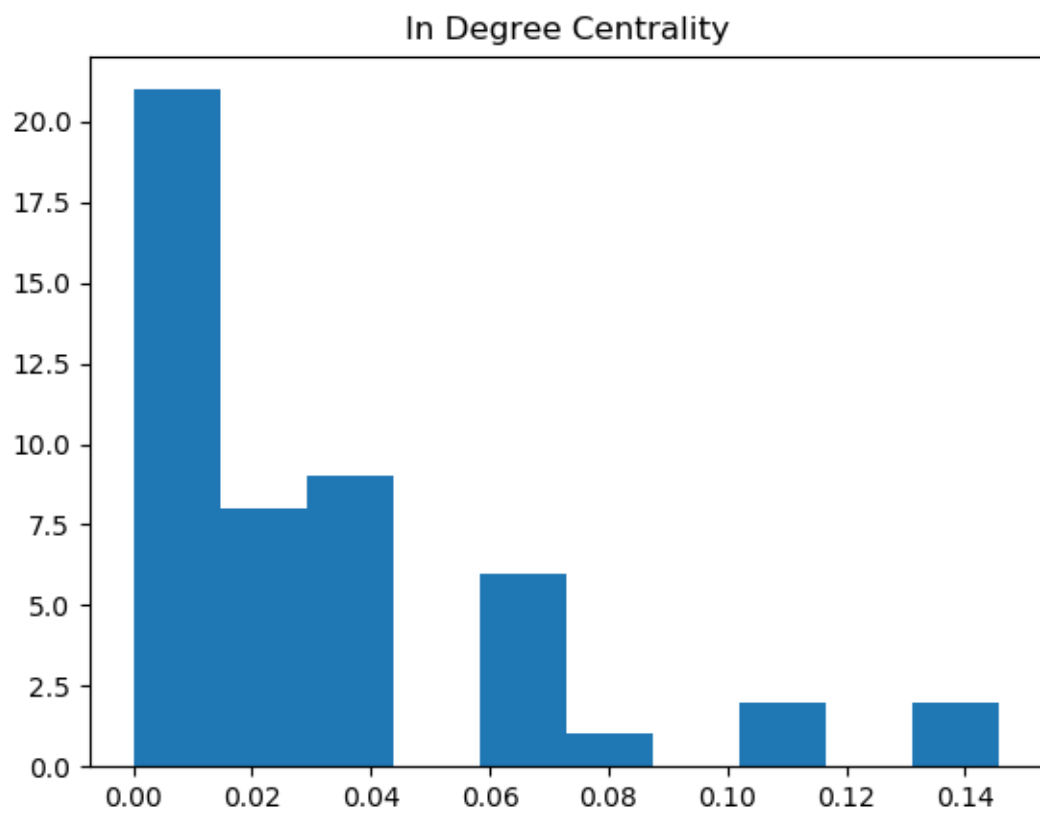


Figure 5: InDegree Centrality

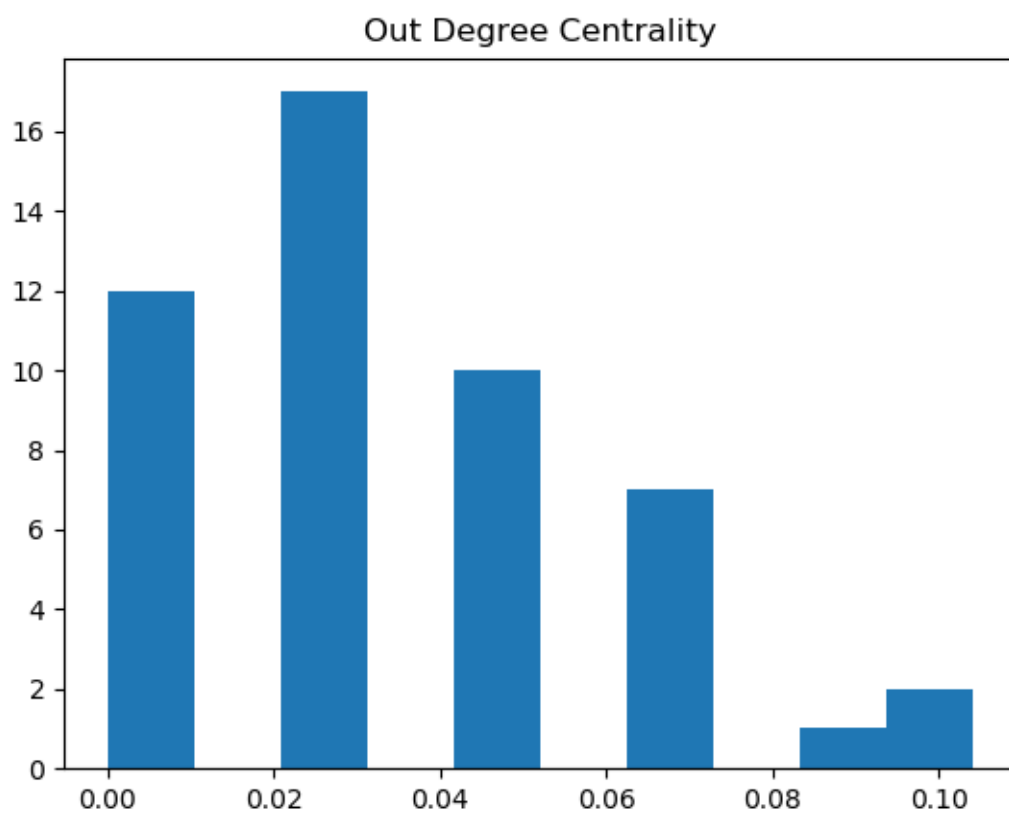


Figure 6: OutDegree Centrality

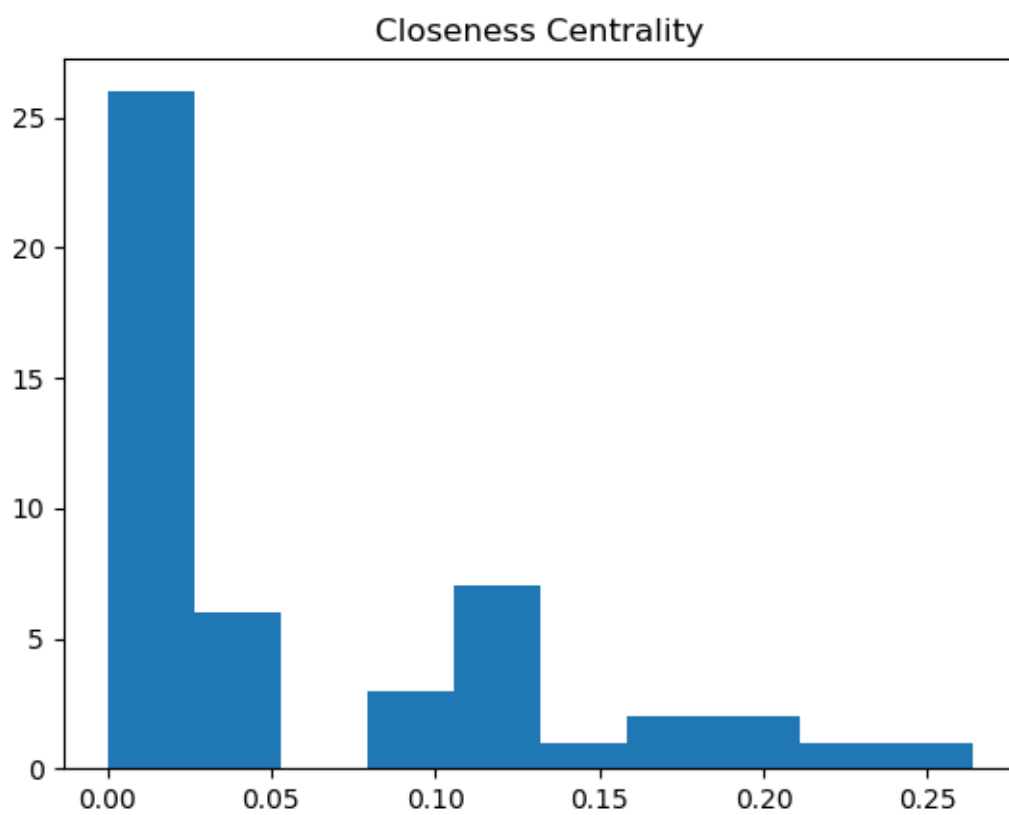


Figure 7: Closeness Centrality

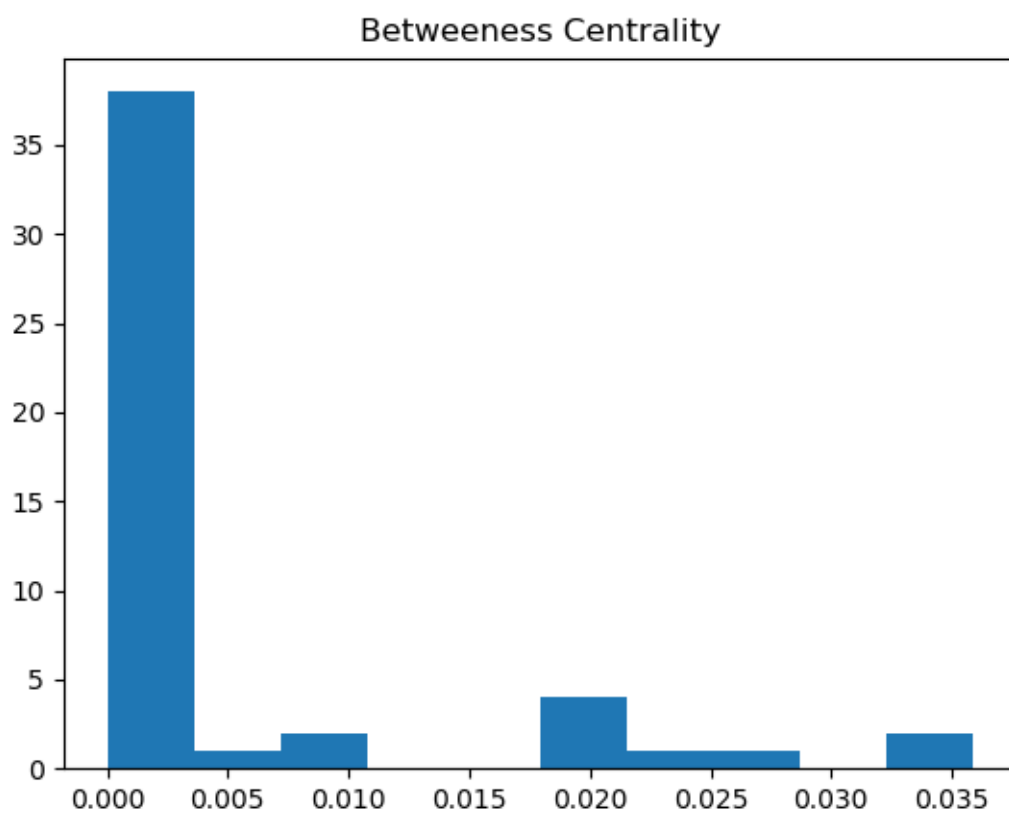


Figure 8: Betweenness Centrality

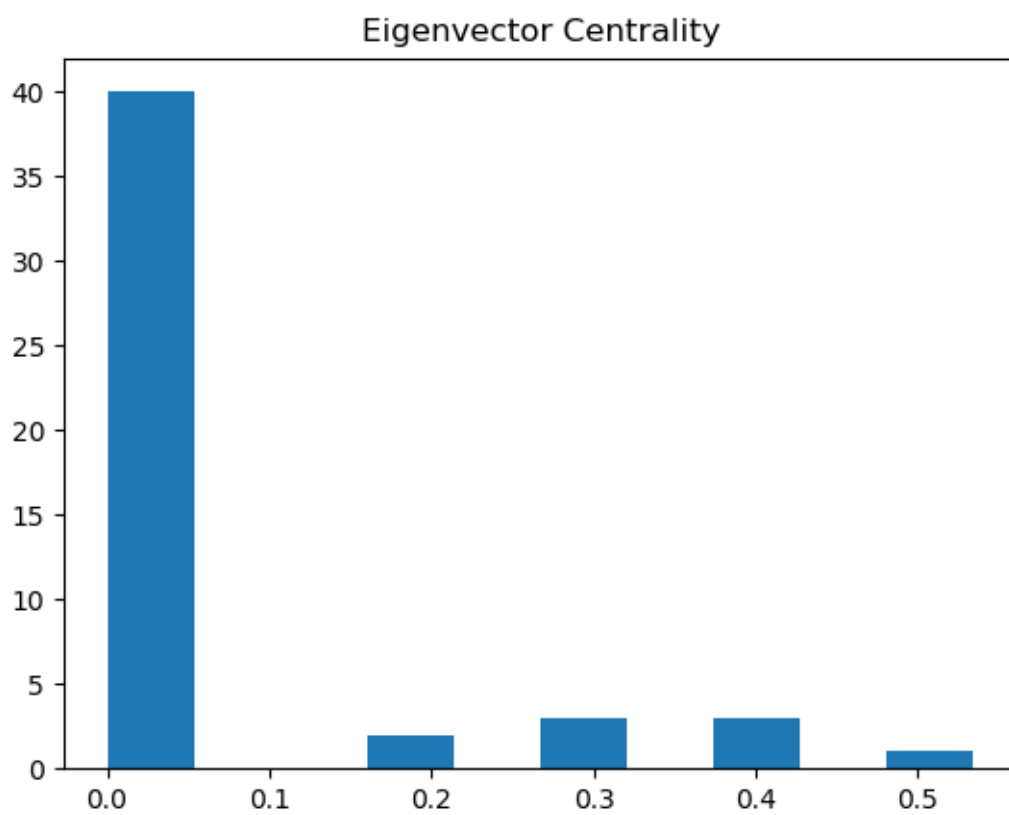


Figure 9: Eigenvector Centrality

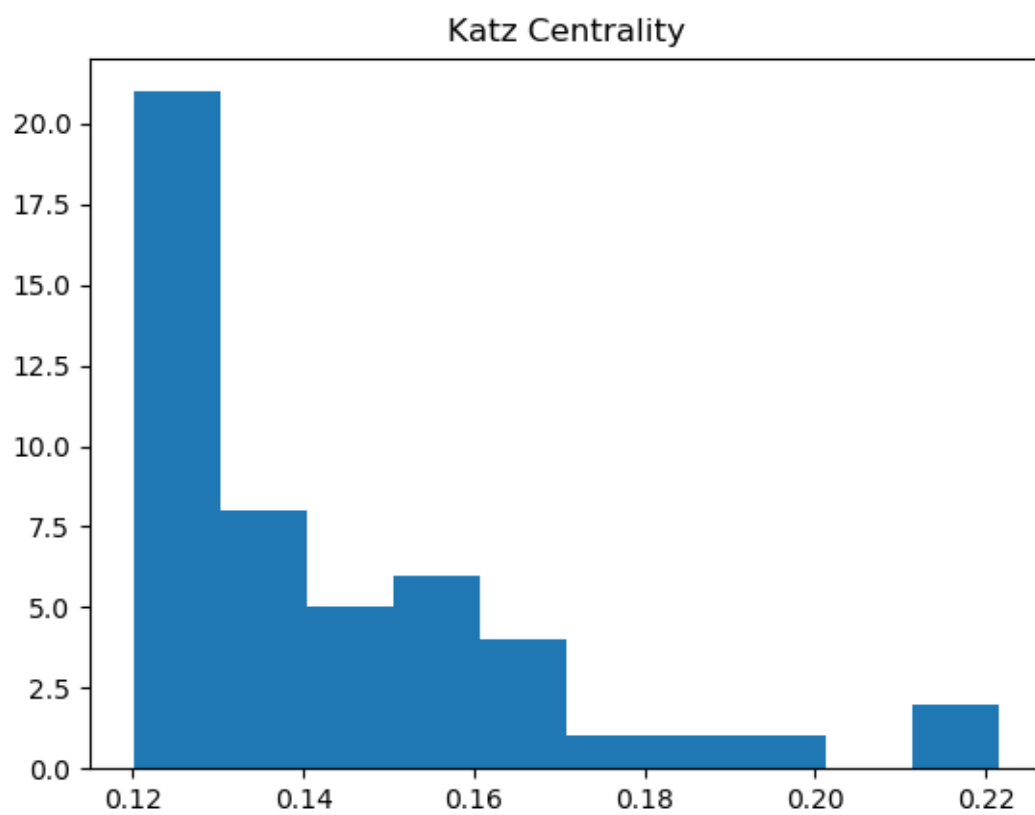


Figure 10: Katz Centrality

2.5 Ερώτημα 5

Στο ερώτημα αυτό υπολογίζονται οι πίνακες V^* και E^* .

Προκειμένου να υπολογιστούν τα V^* και E^* χρησιμοποιούνται τομές συνόλων.

Συγκεκριμένα, βρίσκονται οι κοινοί κόμβοι και οι κοινοί δεσμοί μεταξύ δύο διαδοχικών στιγμών (έστω i και $i + 1$), και κατόπιν αφαιρούνται από την χρονική στιγμή i .

Ο υπολογισμός των V^* και E^* γίνεται πάλι με for loops.

Πολυπλοκότητα αλγορίθμου: $O(n)$.

2.6 Ερώτημα 6

Στο ερώτημα αυτό καλούμαστε να βρούμε τις βασικές μετρικές ομοιότητας σε κάθε υπογράφημα. Η απόσταση γραφήματος και οι κοινοί γείτονες υπολογίζονται στο χέρι με for loops, ενώ οι υπόλοιπες μετρικές υπολογίζονται με έτοιμες συναρτήσεις που μας παρέχει η βιβλιοθήκη networkx.

Μετρικές ομοιότητας:

Graph Distance: - Length of shortest path between u and v .

Common Neighbors: $|\Gamma(u) \cap \Gamma(v)|$

Jaccard's coefficient: $\frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$

Adamic/Adar: $\sum_{z \in |\Gamma(u) \cap \Gamma(v)|} \left[\frac{1}{\log |\Gamma(z)|} \right]$

Preferential attachment: $|\Gamma(u)| \cdot |\Gamma(v)|$

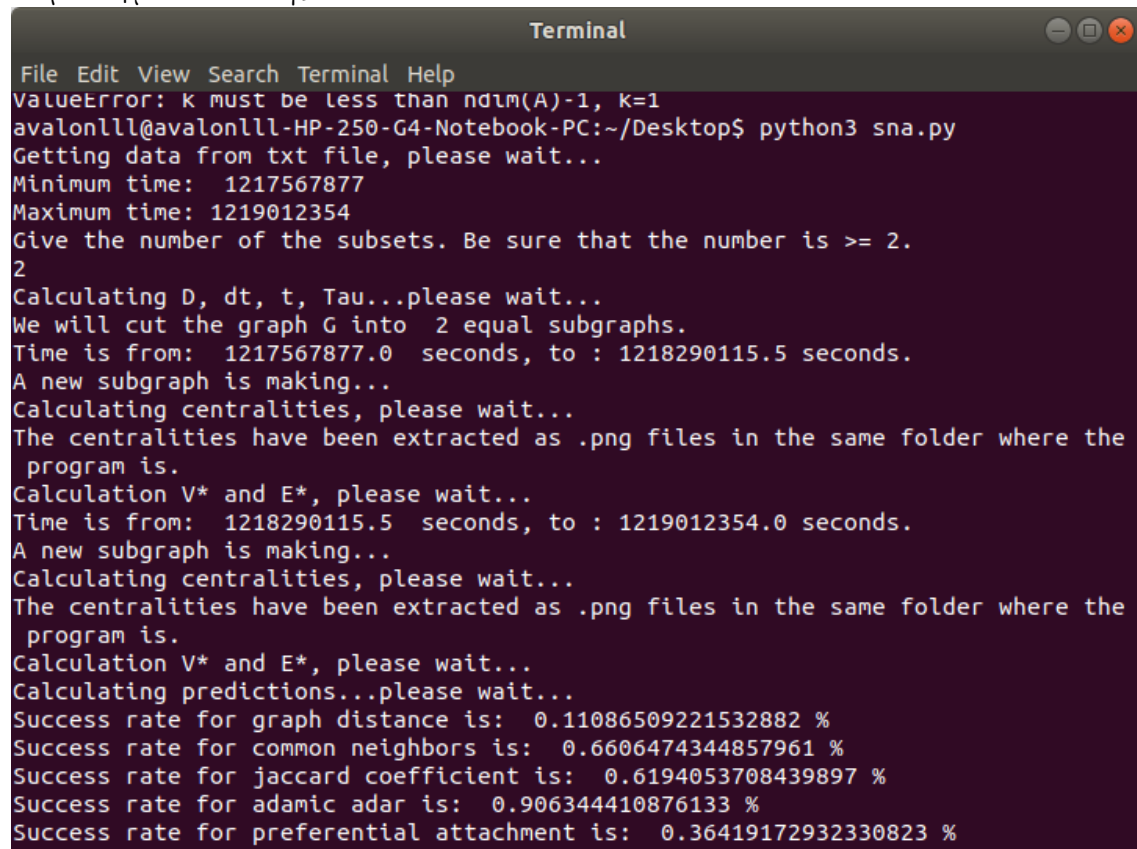
2.7 Ερώτημα 7

Στο ερώτημα αυτό, ο χρήστης δίνει στο πρόγραμμα ένα ποσοστό, από το οποίο ελέγχονται πόσοι δεσμοί προβλέφθηκαν βάσει των μετρικών ομοιότητας που υπολογίστηκαν στο προηγούμενο ερώτημα.

Για παράδειγμα, αν ο χρήστης θέλει να βρει για το καλύτερο 5% πόσοι δεσμοί προβλέφθηκαν και έχουμε 20 αποτελέσματα από τα οποία μόνο 4 προβλέφθηκαν σωστά, το πρόγραμμα θα εμφανίσει ως απάντηση την τιμή της παράστασης $\frac{4}{20}$, ήτοι 20%. Πολυπλοκότητα αλγορίθμου: $O(n)$, αφού η διαδικασία γίνεται μέσα σε ένα for loop με απλούς ελέγχους στο εσωτερικό του.

2.8 Παράδειγμα εκτέλεσης

Παράδειγμα εκτέλεσης:



```
Terminal
File Edit View Search Terminal Help
ValueError: k must be less than ndim(A)-1, k=1
avalonlll@avalonlll-HP-250-G4-Notebook-PC:~/Desktop$ python3 sna.py
Getting data from txt file, please wait...
Minimum time: 1217567877
Maximum time: 1219012354
Give the number of the subsets. Be sure that the number is >= 2.
2
Calculating D, dt, t, Tau...please wait...
We will cut the graph G into 2 equal subgraphs.
Time is from: 1217567877.0 seconds, to : 1218290115.5 seconds.
A new subgraph is making...
Calculating centralities, please wait...
The centralities have been extracted as .png files in the same folder where the
program is.
Calculation V* and E*, please wait...
Time is from: 1218290115.5 seconds, to : 1219012354.0 seconds.
A new subgraph is making...
Calculating centralities, please wait...
The centralities have been extracted as .png files in the same folder where the
program is.
Calculation V* and E*, please wait...
Calculating predictions...please wait...
Success rate for graph distance is: 0.11086509221532882 %
Success rate for common neighbors is: 0.6606474344857961 %
Success rate for jaccard coefficient is: 0.6194053708439897 %
Success rate for adamic adar is: 0.906344410876133 %
Success rate for preferential attachment is: 0.36419172932330823 %
```

Figure 11: Παράδειγμα εκτέλεσης του προγράμματος