Αν. Καθηγητής Π. Λουρίδας

Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

Οικονομικό Πανεπιστήμιο Αθηνών

Οι Αλγόριθμοι στη Χρονιά της Επιδημίας

Ενώ χρησιμοποιούμε τους αλγόριθμους για να λύνουμε προβλήματα με τους υπολογιστές, τα προβλήματα αυτά δεν περιορίζονται στο πεδίο της πληροφορικής. Μάλιστα, πολλές φορές μπορεί ο ίδιος αλγόριθμος να βρίσκει πολύ διαφορετικές εφαρμογές. Για παράδειγμα, οι γράφοι χρησιμοποιούνται για να αναπαραστήσουμε δίκτυα, αλλά δεν υπάρχει περιορισμός στο τι είδους δίκτυα είναι αυτά. Μπορεί να είναι δίκτυα υπολογιστών, διανομής ρεύματος, οδικά δίκτυα, ή κοινωνικά δίκτυα.

Τα κοινωνικά δίκτυα μπορούμε να τα χρησιμοποιήσουμε για να μελετήσουμε τη μετάδοση μιας επιδημίας, αφού συχνά η επιδημία μεταδίδεται μέσω των επαφών μας. Επιπλέον, μπορούμε να χρησιμοποιήσουμε κοινωνικά δίκτυα για να διερευνήσουμε τρόπους αντιμετώπισης μιας επιδημίας. Έτσι, αν έχουμε ένα δίκτυο στο οποίο οι κόμβοι είναι οι άνθρωποι που συνδέονται με άλλους ανθρώπους στους οποίους μπορούν να μεταδώσουν μία νόσο, μπορούμε να αποτρέψουμε μια επιδημία αν εντοπίσουμε ποιους ανθρώπους πρέπει να εμβολιάσουμε ή να απομονώσουμε ώστε πλέον η ασθένεια να μην μπορεί να μεταδοθεί.

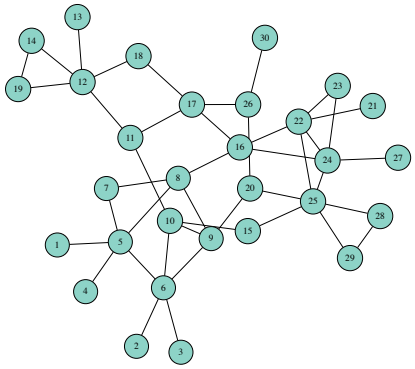
Σημειώστε ότι όταν μιλάμε για επιδημία, δεν είναι απαραίτητο ότι αναφερόμαστε σε βιολογική νόσο. Μπορεί, για παράδειγμα, να μας ενδιαφέρει να σταματήσουμε τη μετάδοση ψευδών ειδήσεων (fake news) μέσα από ένα κοινωνικό δίκτυο, ή τη μετάδοση ενός ιού υπολογιστών, μέσα σε ένα δίκτυο υπολογιστών.

Για τη βέλτιστη ανοσοποίηση ενός δικτύου πρέπει λοιπόν να εντοπίσουμε τους κόμβους εκείνους οι οποίοι έχουν τη μεγαλύτερη επιρροή στο συνολικό δίκτυο. Ένας τρόπος είναι να εργαστούμε ως εξής:

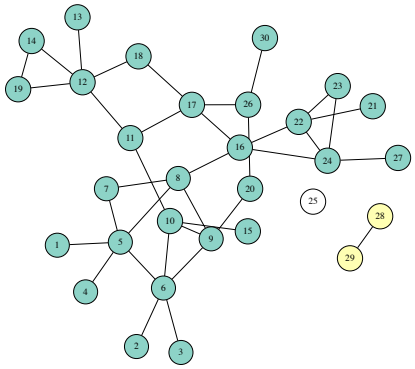
1. Εντοπίζουμε τον κόμβο με το μεγαλύτερο αριθμό συνδέσμων· δηλαδή, τον κόμβο με τον μεγαλύτερο *βαθμό* (degree). Σε περίπτωση που υπάρχουν περισσότεροι από έναν κόμβοι με τον ίδιο, μέγιστο, αριθμό συνδέσμων, επιλέγουμε αυτόν με το μικρότερο αριθμητικά όνομα.
2. Εμβολιάζουμε ή απομονώνουμε, δηλαδή αφαιρούμε, αυτόν τον κόμβο (τον αφαιρούμε αφού πλέον δεν μπορεί ο συγκεκριμένος κόμβος να μεταδώσει ασθένεια).
3. Επιστρέφουμε στο βήμα 1.

Επαναλαμβάνουμε τα βήματα 1–3 για όσους κόμβους θέλουμε.

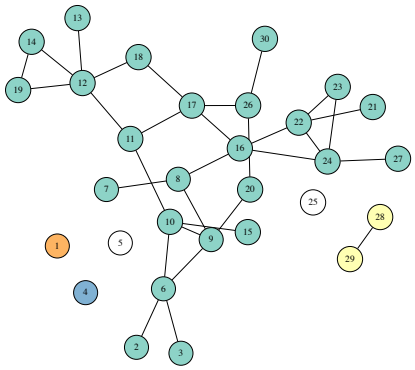
Στις παρακάτω εικόνες μπορείτε να δείτε πώς προχωράμε στον εμβολιασμό ενός δικτύου αφαιρώντας τους τέσσερεις κόμβους με τον μεγαλύτερο κάθε φορά αριθμό συνδέσμων. Με λευκό χρωματίζονται οι κόμβοι που αφαιρούνται διαδοχικά, ενώ χρησιμοποιούμε άλλα χρώματα για τα συνδεδεμένα κομμάτια του γράφου.



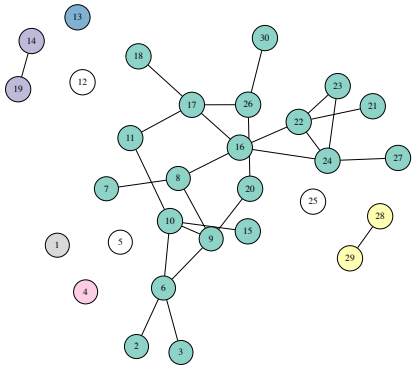
Σχήμα 1: Αρχικό Δίκτυο



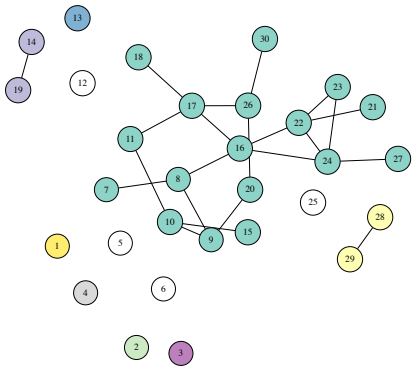
Σχήμα 2: Αφαίρεση 1ου κόμβου



Σχήμα 3: Αφαίρεση 2ου κόμβου



Σχήμα 4: Αφαίρεση 3ου κόμβου



Σχήμα 5: Αφαίρεση 4ου κόμβου

Με τον τρόπο αυτό, βλέπουμε ότι μετά από την αφαίρεση τεσσάρων κόμβων το μεγαλύτερο *συνδεδεμένο κομμάτι* (connected component) του δικτύου περιέχει 17 κόμβους, άρα αν ασθενίσει κάποιος κόμβος εκεί, η νόσος θα μεταδωθεί σε άλλους 16 κόμβους.

Ένας άλλος τρόπος εργασίας είναι σε πολλά δίκτυα πιο αποτελεσματικός. Αντί να θεωρούμε ότι ο κόμβος με τη μεγαλύτερη επιρροή είναι αυτός με τους περισσότερους συνδέσμους, ορίζουμε τη *συνολική επιρροή* (collective influence) ενός κόμβου:

$$\mathrm{CI}(i, r) = (k_i - 1) \sum_{j \in \partial \mathrm{Ball}(i,r)} (k_j - 1)$$

Στον παραπάνω ορισμό:

- i είναι ο κόμβος του οποίου υπολογίζουμε τη συνολική επιρροή και k_i είναι ο αριθμός συνδέσμων του κόμβου i.
- Με $\mathrm{Ball}(i, r)$ συμβολίζουμε το σύνολο των κόμβων των οποίων το συντομότερο μονοπάτι από τον κόμβο i έχει μήκος μικρότερο ή ίσο του r. Αν μπορούσατε να ζωγραφίσετε έναν κύκλο με ακτίνα r συνδέσμους γύρω από τον κόμβο i, οι κόμβοι j είναι οι κόμβοι που θα πέφτανε μέσα στον κύκλο. Αν προτιμούμε να μιλάμε σε τρεις διαστάσεις, είναι οι κόμβοι οι οποίοι βρίσκονται στην επιφάνεια μιας σφαίρας με κέντρο τον κόμβο i και ακτίνα r.
- Με $\partial\mathrm{Ball}(i, r)$ συμβολίζουμε το σύνολο των κόμβων των οποίων το συντομότερο μονοπάτι από τον κόμβο i έχει μήκος ακριβώς r. Αν μπορούσατε να ζωγραφίσετε έναν κύκλο με ακτίνα r συνδέσμους γύρω από τον κόμβο i, οι κόμβοι j είναι οι κόμβοι που θα πέφτανε πάνω στην περιφέρεια του κύκλου. Αν προτιμούμε να μιλάμε σε τρεις διαστάσεις, είναι οι κόμβοι οι οποίοι βρίσκονται στην επιφάνεια μιας σφαίρας με κέντρο τον κόμβο i και ακτίνα r.

Σύμφωνα με τα παραπάνω, για να υπολογίσουμε τη συνολική επιρροή ενός κόμβου,

βρίσκουμε τους κόμβους που το συντομότερο μονοπάτι τους από τον κόμβο i είναι ίσο με r και σχηματίζουμε το άθροισμα που προκύπτει από τον αριθμό συνδέσμων του καθενός, j, από αυτούς τους κόμβους μείον ένα. Στη συνέχεια πολλαπλασιάζουμε το αποτέλεσμα με τον αριθμό συνδέσμων του κόμβου i μείον ένα.

Συνεπώς για να εμβολιάσουμε ένα δίκτυο χρησιμοποιώντας τη συνολική επιρροή, εργαζόμαστε ως εξής:

1. Υπολογίζουμε τη συνολική επιρροή του κάθε κόμβου.
2. Επιλέγουμε τον κόμβο με τη μεγαλύτερη συνολική επιρροή. Σε περίπτωση που υπάρχουν περισσότεροι από έναν κόμβοι με την ίδια, μέγιστη, συνολική επιρροή, επιλέγουμε αυτόν με το μικρότερο αριθμητικά όνομα.
3. Εμβολιάζουμε (αφαιρούμε) αυτόν τον κόμβο.
4. Ενημερώνουμε τη συνολική επιρροή των κόμβων που επηρεάζονται από την αφαίρεση του κόμβου στο βήμα 3. Αυτοί είναι οι κόμβοι $\text{Ball}(i, r+1)$, δηλαδή όσοι βρίσκονται μέσα στον κύκλο (ή στη σφαίρα) με ακτίνα $r+1$ συνδέσμους από τον κόμβο i, όπου i είναι ο κόμβος που αφαιρέσαμε στο βήμα 3.
5. Επιστρέφουμε στο βήμα 2.

Επαναλαμβάνουμε τα βήματα 2–5 για όσους κόμβους θέλουμε.

Για να υλοποιήσουμε τον αλγόριθμο πρέπει να έχουμε έναν τρόπο να υπολογίζουμε το $\partial\text{Ball}(i, r)$ και το $\text{Ball}(i, r+1)$. Για να εντοπίσουμε τους κόμβους που βρίσκονται μέχρι μια συγκεκριμένη απόσταση από έναν κόμβο, αρκεί μια παραλλαγή της κατά πλάτος αναζήτησης (breadth-first search, BFS), όπου, καθώς απομακρυνόμαστε από τον κόμβο εκκίνησης μετράμε την απόσταση από αυτόν.

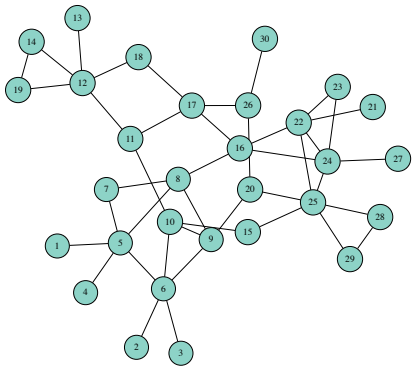
Στις παρακάτω εικόνες μπορείτε να δείτε πώς προστατεύουμε ένα δίκτυο αφαιρώντας τους τέσσερεις κόμβους με τη μεγαλύτερη κάθε φορά συνολική επιρροή, έχοντας $r = 2$. Και πάλι, με λευκό χρωματίζονται οι κόμβοι που αφαιρούνται διαδοχικά, με άλλα χρώματα τα συνδεδεμένα κομμάτια.

Τώρα βλέπουμε ότι μετά από την αφαίρεση ήδη τριών κόμβων, το μεγαλύτερο συνδεδεμένο κομμάτι του δικτύου περιέχει εννέα μόνο κόμβους. Άρα ήδη με τρεις κόμβους έχουμε επιτύχει καλύτερη καταστροφή από ό,τι με τέσσερεις κόμβους με την προηγούμενη μέθοδο. Ο κόμβος 10 αφαιρείται πρώτος καθώς έχει τη μεγαλύτερη επιρροή, ίση με 63. Επιβεβαιώστε τον υπολογισμό αυτό ώστε να καταλάβετε τον ορισμό της συνολικής επιρροής.

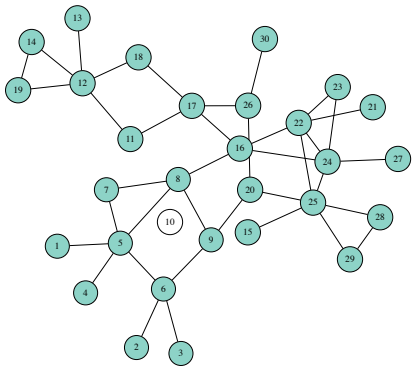
Σκοπός της εργασίας είναι η κατασκευή ενός προγράμματος σε γλώσσα Python 3 το οποίο θα μπορεί να διαβάζει ένα αρχείο που περιγράφει ένα γράφο και να επιλέγει τους κόμβους που θα αφαιρέσει με τους δύο αυτούς τρόπους.

Απαιτήσεις Προγράμματος

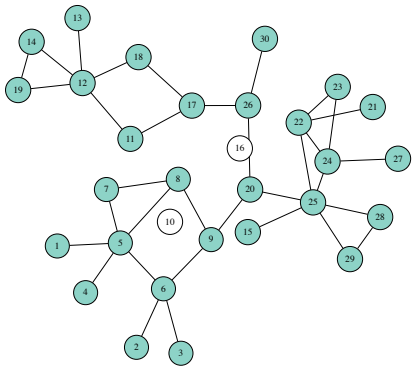
Κάθε φοιτητής θα εργαστεί σε αποθετήριο στο GitHub. Για να αξιολογηθεί μια εργασία θα πρέπει να πληροί τις παρακάτω προϋποθέσεις:



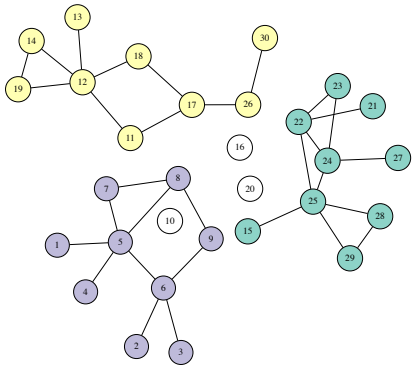
Σχήμα 6: Αρχικό Δίκτυο



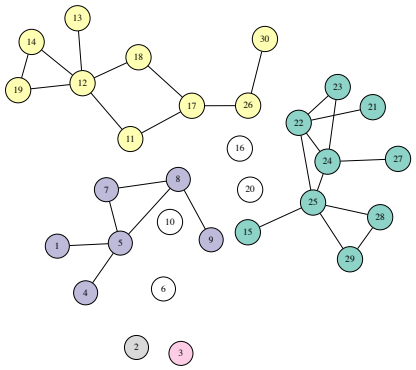
Σχήμα 7: Αφαίρεση 1ου κόμβου



Σχήμα 8: Αφαίρεση 2ου κόμβου



Σχήμα 9: Αφαίρεση 3ου κόμβου



Σχήμα 10: Αφαίρεση 4ου κόμβου

- Για την υποβολή της εργασίας θα χρησιμοποιηθεί το ιδιωτικό αποθετήριο του φοιτητή που δημιουργήθηκε για τις ανάγκες του μαθήματος και του έχει αποδωθεί. Το αποθετήριο αυτό έχει όνομα του τύπου `username-algo-assignments`, όπου `username` είναι το όνομα του φοιτητή στο GitHub. Για παράδειγμα, το σχετικό αποθετήριο του διδάσκοντα θα ονομαζόταν `louridas-algo-assignments` και θα ήταν προσβάσιμο στο https://github.com/dmst-algorithms-course/louridas-algo-assignments. *Τυχόν άλλα αποθετήρια απλώς θα αγνοηθούν.*
- Μέσα στο αποθετήριο αυτό θα πρέπει να δημιουργηθεί ένας κατάλογος `assignment-2020-2`.
- Μέσα στον παραπάνω κατάλογο το πρόγραμμα θα πρέπει να αποθηκευτεί με το όνομα `network_destruction.py`.
- Για την υλοποίηση του γράφου θα πρέπει να χρησιμοποιήσετε λίστες γειτνίασης (adjacency lists) και όχι πίνακα γειτνίασης (adjacency matrix).
- Δεν επιτρέπεται η χρήση έτοιμων βιβλιοθηκών γράφων ή τυχόν έτοιμων υλοποιήσεων των αλγορίθμων, ή τμημάτων αυτών, εκτός αν αναφέρεται ρητά ότι επιτρέπεται.
- Επιτρέπεται η χρήση δομών δεδομένων της Python όπως στοίβες, λεξικά, σύνολα, κ.λπ.
- Επιτρέπεται η χρήση της βιβλιοθήκης `argparse` ή της βιβλιοθήκης `sys` (συγκεκριμένα, της λίστας `sys.argv`) προκειμένου να διαβάσει το πρόγραμμα τις παραμέτρους εισόδου.
- Το πρόγραμμα θα πρέπει να είναι γραμμένο σε Python 3.

Κλήση Προγράμματος

Το πρόγραμμα θα μπορεί να καλείται ως εξής:

```
python network_destruction.py [-c] [-r RADIUS] num_nodes input_file
```

Αναλόγως του λειτουργικού συστήματος του υπολογιστή σας μπορεί να πρέπει να δώσετε `python3` ή κάτι άλλο αντί για `python`.

Οι αγκύλες δεν αποτελούν μέρος των πραγμάτων που δίνει ο χρήστης, απλώς χρησιμεύουν για να μας υπενθυμίζουν στην περιγραφή του προγράμματος ότι η συγκεκριμένη παράμετρος είναι προαιρετική. Έτσι έχουμε:

- Αν δίνεται η παράμετρος `-c`, το πρόγραμμα θα χρησιμοποιεί τον αριθμό συνδέσμων κάθε κόμβου και όχι τη συνολική επιρροή. Δηλαδή, θα λειτουργεί όπως το πρώτο παράδειγμα που δώσαμε. Διαφορετικά, το πρόγραμμα θα χρησιμοποιεί τη συνολική επιρροή, δηλαδή θα λειτουργεί όπως το δεύτερο παράδειγμα.
- Αν δίνεται η παράμετρος `-r RADIUS` το πρόγραμμα θα χρησιμοποιεί την τιμή `RADIUS` για το r.
- Η παράμετρος `num_nodes` αντιστοιχεί στον αριθμό των κόμβων που θέλουμε να αφαιρέσουμε.
- Η παράμετρος `input_file` είναι το όνομα του αρχείου που περιγράφει τον γράφο.

Το αρχείο `input_file` θα είναι της μορφής:

```
1 2
1 3
2 4
...
```

δηλαδή αποτελείται από γραμμές που η κάθε μία περιέχει δύο αριθμούς. Αν οι δύο αριθμοί είναι οι `x` και `y`, ο γράφος θα έχει ένα σύνδεσμο μεταξύ των κόμβων `x` και `y`. Ο γράφος δεν θα είναι κατευθυνόμενος, άρα θα θεωρούμε πάντα ότι θα υπάρχει και ο αντίστροφος σύνδεσμος από το `y` στο `x`. Οι κόμβοι θα είναι πάντα αριθμοί και θα είναι 1, 2, ...

Η έξοδος του προγράμματος θα εμφανίζει σε μία σειρά γραμμών τον κάθε κόμβο που αφαιρεί και τη μετρική του. *Προσοχή:* αν η έξοδος δεν είναι ακριβώς σύμφωνη με την περιγραφή της, η εργασία *δεν θα μπορεί να αξιολογηθεί.*

Παραδείγματα Εκτέλεσης

Παράδειγμα 1

Αν χρησιμοποιήσουμε το αρχείο υπόδειγμα destruction_example_1.txt ως εξής:

```
python network_destruction.py -r 2 4 destruction_example_1.txt
```

θα πάρουμε ακριβώς:

```
10 63
16 51
20 30
6 9
```

Δηλαδή πρώτα αφαιρέσαμε τον κόμβο 10 με συνολική επιρροή 63, στη συνέχεια τον κόμβο 16 με συνολική επιρροή 51, κ.λπ.

Αν χρησιμοποιήσουμε το αρχείο υπόδειγμα destruction_example_1.txt ως εξής:

```
python network_destruction.py -c 4 destruction_example_1.txt
```

θα πάρουμε ακριβώς:

```
25 6
5 5
12 5
6 4
```

Τώρα πρώτα αφαιρέσαμε τον κόμβο 25 με βαθμό 6, στη συνέχεια τον κόμβο 5 που έχει βαθμό 5, κ.ο.κ.

Παράδειγμα 2

Αν χρησιμοποιήσουμε το αρχείο υπόδειγμα destruction_example_2.txt ως εξής:

```
python network-destruction-optimised.py -c 6 destruction_example_2.txt
```

θα πάρουμε ακριβώς:

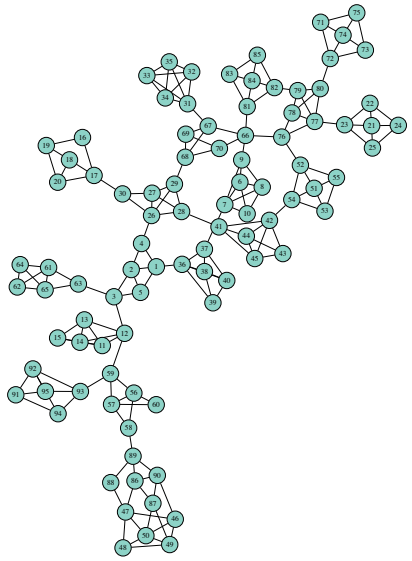
```
7 5
12 5
26 5
31 5
36 5
42 5
```

Αν δώσουμε:

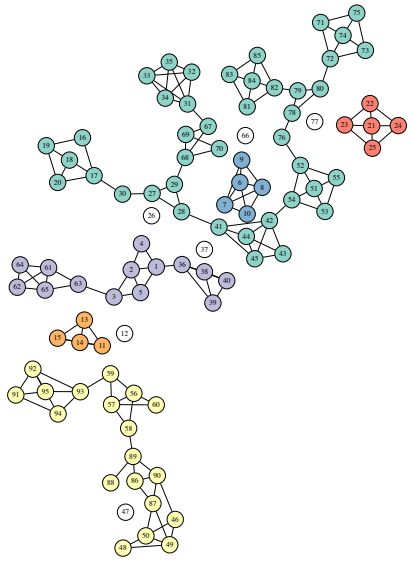
```
python network_destruction.py -r 2 6 destruction_example_2.txt
```

θα πάρουμε ακριβώς (βλ. στην επόμενη σελίδα):

```
66 148
12 72
77 64
26 60
47 48
37 45
```



Σχήμα 11: Δεύτερο Παράδειγμα, Αρχική Κατάσταση



Σχήμα 12: Δεύτερο Παράδειγμα, Τελική Κατάσταση

Παράδειγμα 3

Αν χρησιμοποιήσουμε το αρχείο υπόδειγμα destruction_example_3.txt ως εξής:

```
python network-destruction-optimised.py -c 10 destruction_example_3.txt
```

θα πάρουμε ακριβώς:

```
114 7
105 6
112 6
125 6
6 4
23 4
34 4
65 4
107 4
124 4
```

Αν δώσουμε:

```
python network_destruction.py -r 3 10 destruction_example_3.txt
```

θα πάρουμε ακριβώς (βλ. στην επόμενη σελίδα):

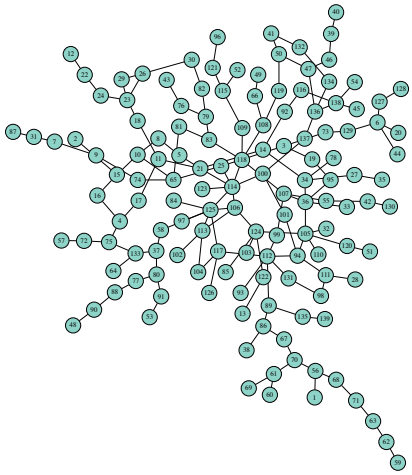
```
114 270
105 125
112 75
118 39
125 25
11 24
65 21
138 21
36 15
37 8
```

Προαιρετικά

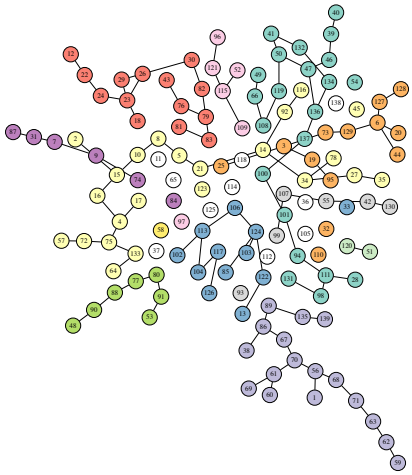
Για να καταλάβουμε πώς λειτουργεί ο αλγόριθμος, μια καλή ιδέα είναι να προσθέσετε μία ακόμα παράμετρο στο πρόγραμμα, παράδειγμα `-t` (trace), με την οποία μετά από κάθε αφαίρεση κόμβου θα μας εμφανίζει τα συνδεδεμένα τμήματα του γράφου.

Από εκεί και μετά, αν υπάρχει αυτή η λειτουργικότητα, είναι δυνατή και η οπτικοποίηση της εξέλιξης, προσθέτοντας στο πρόγραμμα τη δυνατότητα να δημιουργεί μια σειρά εικόνων όπως αυτές που βλέπετε στην παρούσα εκφώνηση. Για τη δημιουργία τους χρησιμοποιήθηκε η βιβλιοθήκη graphviz μέσω της διεπαφής της με Python.

Καλή Επιτυχία.



Σχήμα 13: Τρίτο Παράδειγμα, Αρχική Κατάσταση



Σχήμα 14: Τρίτο Παράδειγμα, Τελική Κατάσταση

Για Περισσότερες Πληροφορίες

- István A. Kovács and Albert-László Barabási, Destruction perfected, Nature 524, 38–39 (06 August 2015). doi:10.1038/524038a
- Flaviano Morone and Hernán A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524, 65–68 (06 August 2015). doi:10.1038/nature14604
- Flaviano Morone, Byungjoon Min, Lin Bo, Romain Mari & Hernán A. Makse, Collective Influence Algorithm to find influencers via optimal percolation in massively large social media, Scientific Reports, 6:30062 (26 July 2016). doi:10.1038/srep30062

Επιπλέον, την περίοδο αυτή που δεν μπορούμε να βγούμε από τα σπίτια μας παρά μόνο για συγκεκριμένους λόγους, είναι ευκαιρία να γνωρίσετε, αν δεν τα έχετε γνωρίσει ακόμα, μερικά επίκαιρα αναγνώσματα:

- Η Πανούκλα, Albert Camus.
- Περί Τυφλότητος, José Saramago.
- Σταθμός 11, Emily St. John Mandel.