2017-2018

Θέμα: Υλοποίηση & απεικόνιση BFS αλγορίθμου με χρήση βιβλιοθήκης   
NetworkX και Matplotlib.

ομαδα 37

Η ομάδα 37 με υπεύθυνο καθηγητή τον κύριο Παναγιώτη Ντίλιο, απαρτίζεται από τους φοιτητές Φωτεινή Άρτεμις Κισκήρα (up 1059448), Σοφία Μάμαλη (up1059466), Μιχάλη Μαναγούδη (up1059398), Αντρέα Μάρκου (up1056580), Ηλία Ματζαράπη(1063966), Νίκο Παπανικολάου(up1059516) και Άννα Σεραφείμ (up1064481).

Μόλις μας δόθηκε το θέμα, ( Υλοποίηση και απεικόνιση αλγορίθμου ΒFS με χρήση της βιβλιοθήκης Matplotlib και NetworkX) απευθυνθήκαμε στον υπεύθυνο καθηγητή με σκοπό τη καλύτερη κατανόηση αυτού. Αφού μας έστειλε αρκετό βοηθητικό υλικό στο eclass, το αξιοποιήσαμε μελετώντας ο καθένας μόνος του και βρίσκοντας παράλληλα πρόσθετες πληροφορίες. Στη συνέχεια, αρχίσαμε τις καθημερινές συναντήσεις όπου συζητήσαμε το θέμα ενδελεχώς, καταλήγοντας στον τρόπο με τον οποίο θα δουλέψουμε. Ύστερα, αρχίσαμε την ανάπτυξη του κώδικα αφού πρώτα κατεβάσαμε το πρόγραμμα Αnaconda. Μέσα στο διάλειμμα των Χριστουγεννιάτικων διακοπών, επικοινωνήσαμε μέσω skype και σχεδόν ολοκληρώσαμε τον κώδικα ώστε όταν επιστρέφαμε να κάναμε απλώς κάποιες διορθώσεις.

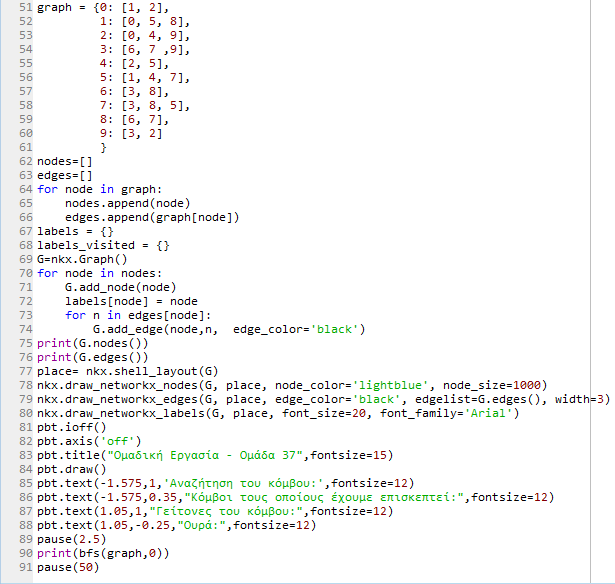
Όλοι μας εργαστήκαμε ισάξια και παραβρεθήκαμε στις συγκεντρώσεις της ομάδας ώστε το αποτέλεσμα που πρόεκυψε να είναι «γέννημα» συλλογικής προσπάθειας. Συγκεκριμένα, χωριστήκαμε σε 3 ομάδες. Η Φωτεινή, ο Ηλίας και ο Μιχάλης ασχολήθηκαν περισσότερο με τη βιβλιοθήκη NetworkX, ο Νίκος με τον Αντρέα με τη βιβλιοθήκη Matplotlib και η Σοφία με την Άννα με τον αλγόριθμο BFS χωρίς ωστόσο να χάσει ο καθένας την επαφή με τα υπόλοιπα μέρη του κώδικα.

Ξεκινώντας τον κώδικα μας, γράψαμε πρώτα τον γράφο (γράφημα) στον οποίο πραγματοποιείται η ένωση του κάθε κόμβου με τους γειτονικούς του. Έπειτα, γράψαμε τον BFS αλγόριθμο, ο όποιος δέχεται σαν είσοδο ένα πεδίο καταστάσεων (κόμβοι), μια αρχική και μια τελική κατάσταση. Η λειτουργιά του αλγόριθμου είναι να βρει το «μονοπάτι» από την αρχική στην τελική θέση επιλέγοντας με τον οποιοδήποτε τρόπο θέλει τις ενδιάμεσες καταστάσεις. Τέλος, χρησιμοποιήσαμε τις βιβλιοθήκες NetworkΧ και Matplotlib με τις οποίες φτιάξαμε τα γραφικά.

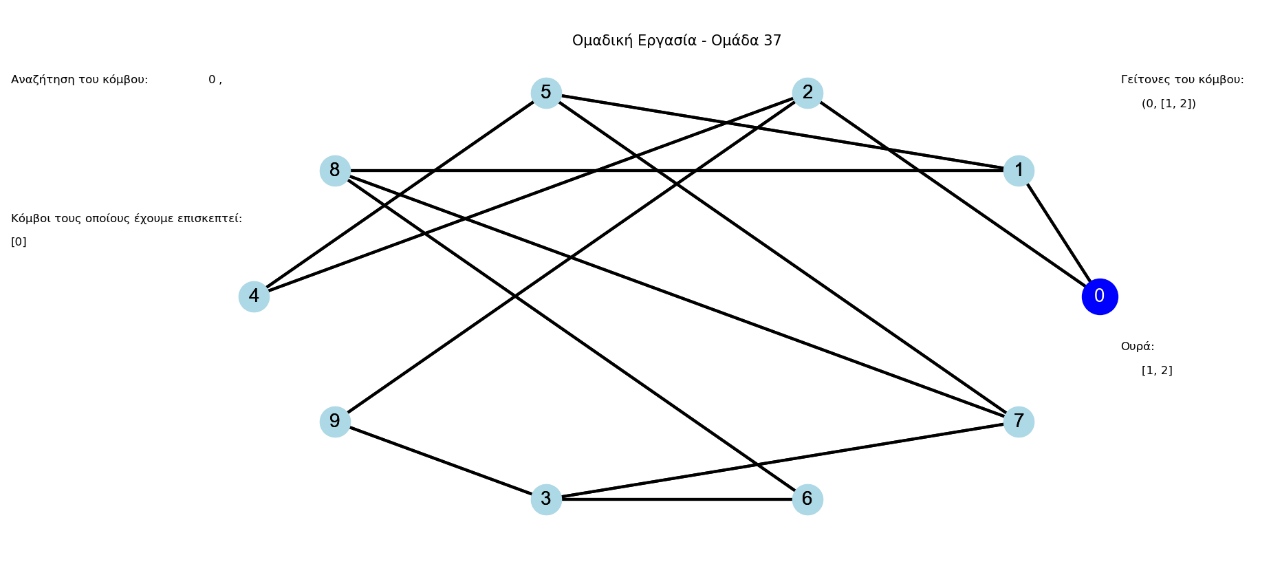
Σε ό,τι αφορά τις οδηγίες εγκατάστασης Anaconda (mini) και Virtual Environments (για Python projects), έχουμε παραθέσει και ένα ξεχωριστό αρχείο σε περίπτωση που δεν έχουν εγκατασταθεί στον υπολογιστή του χρηστή. Συνοπτικά, από το http://conda.io/miniconda.html, κατεβάζουμε την έκδοση που ταιριάζει στο λειτουργικό μας. Έπειτα ,τρέχουμε το αρχείο εγκατάστασης σαν Διαχειριστής ,επιλέγουμε ένα φάκελο για την εγκατάστασή του. Κατόπιν δημιουργώντας ένα εικονικό περιβάλλον κατεβάζουμε τις 2 βιβλιοθήκες που χρειαζόμαστε στην σωστή έκδοση, ώστε να ταιριάζουν στο λειτουργικό μας σύστημα και στην έκδοση της Python που έχουμε εγκατεστημένη στον υπολογιστή μας.

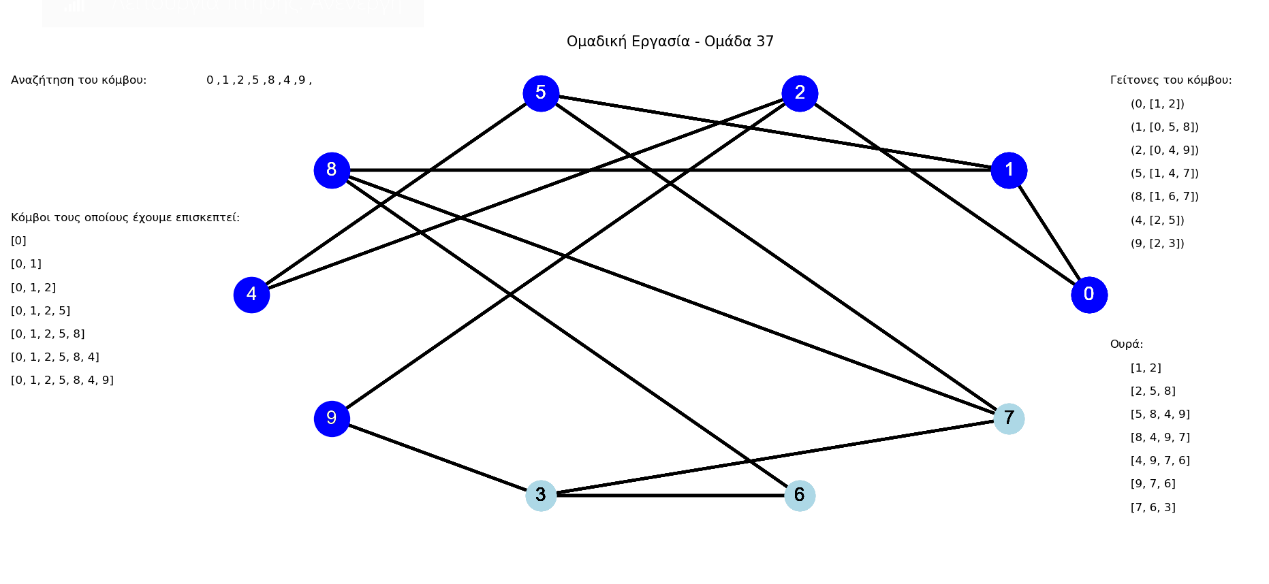
Ο κώδικάς μας αναπτύσσεται στο παρακάτω παράρτημα:

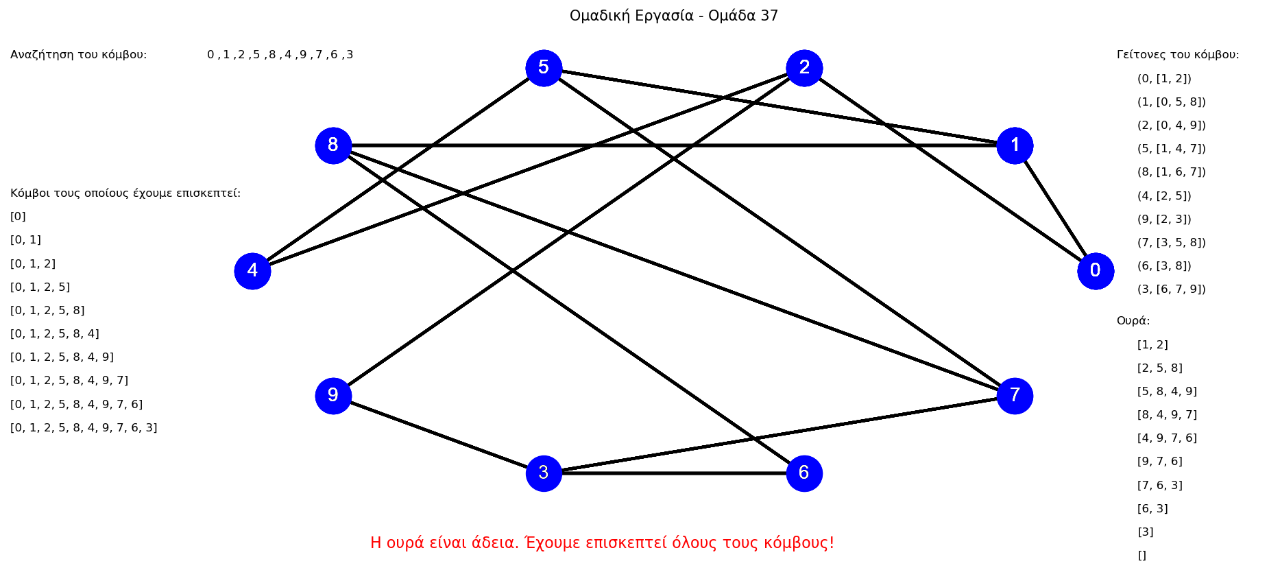




Στην συνέχεια παρατίθενται στιγμιότυπα από την εκτέλεση του προγράμματος μας:







Βιβλιογραφία:

<http://eddmann.com/posts/depth-first-search-and-breadth-first-search-in-python/>

<https://pythoninwonderland.wordpress.com/2017/03/18/how-to-implement-breadth-first-search-in-python/>

<https://www.youtube.com/watch?v=vzjg3T72tRs>

<https://www.youtube.com/watch?v=0Zsabo7ires>

<https://www.youtube.com/watch?v=03GTwWRRx7Y>

<https://matplotlib.org/api/_as_gen/matplotlib.pyplot.title.html?highlight=title#matplotlib.pyplot.>[title](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.title.html?highlight=title)

<https://networkx.github.io/documentation/stable/reference/classes/generated/networkx.Graph.nodes.html?highlight=nodes#networkx.Graph.>[nodes](https://networkx.github.io/documentation/stable/reference/classes/generated/networkx.Graph.nodes.html?highlight=nodes)

<https://networkx.github.io/documentation/stable/reference/classes/generated/networkx.Graph.edges.html?highlight=edges#networkx.Graph.>[edges](https://networkx.github.io/documentation/stable/reference/classes/generated/networkx.Graph.edges.html?highlight=edges)

<https://networkx.github.io/documentation/stable/auto_examples/drawing/plot_labels_and_colors.html?highlight=>[nodelist](https://networkx.github.io/documentation/stable/auto_examples/drawing/plot_labels_and_colors.html?highlight=nodelist)

<https://networkx.github.io/documentation/stable/reference/generated/networkx.drawing.nx_pylab.draw_networkx.html?highlight=>[nodelist](https://networkx.github.io/documentation/stable/reference/generated/networkx.drawing.nx_pylab.draw_networkx.html?highlight=nodelist)

<https://matplotlib.org/api/_as_gen/matplotlib.pyplot.text.html?highlight=text#matplotlib.pyplot.>[text](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.text.html?highlight=text)

<https://matplotlib.org/api/_as_gen/matplotlib.pyplot.axes.html?highlight=axes#matplotlib.pyplot.>[axes](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.axes.html?highlight=axes)

<https://www.linkedin.com/pulse/graphs-python-dfs-henrique-gabriel-gularte-pereira/>

<https://medium.com/basecs/from-theory-to-practice-representing-graphs-cfd782c5be38>

<https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/a/breadth-first-search-and-its-uses>

<https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/a/route-finding>