



ΑΛΓΟΡΙΘΜΟΙ

Διδάσκουσα: Κ. Παπακωνσταντινοπούλου

3η Σειρά Προγραμματιστικών και Γραπτών Ασκήσεων Ημ/νία παράδοσης: 26/5/2019

Οι λύσεις των προγραμματιστικών ασκήσεων πρέπει να υλοποιηθούν σε γλώσσα προγραμματισμού **Java** και τα **αρχεία** σας πρέπει να υποβληθούν στο **eClass** σε ένα συμπίεσμένο φάκελο .zip με όνομα τον αριθμό μητρώου σας, δηλαδή 3xxxxxx.zip. Τις λύσεις των γραπτών ασκήσεων πρέπει να τις γράψετε σε ηλεκτρονική μορφή και να τις συμπεριλάβετε σε ένα pdf αρχείο μέσα στον παραπάνω φάκελο.

Προγραμματιστικές Ασκήσεις

Άσκηση 3.1

Δίνεται ένα σύνολο δεδομένων που παριστάνει ένα μη κατευθυνόμενο δίκτυο φιλοσόφων, με συνδέσεις μεταξύ φιλοσόφων των οποίων το έργο σχετίζεται. Το δίκτυο παριστάνεται με ζεύγη κόμβων. Το αρχείο με τα δεδομένα βρίσκεται στη διεύθυνση: https://eclass.aueb.gr/modules/document/file.php/INF161/Datasets/philosophy_edgelist.txt και μπορείτε να το ανοίξετε αφού κάνετε login στο eClass.

Να γράψετε πρόγραμμα Java που παίρνει ως είσοδο δύο φιλοσόφους και βρίσκει το συντομότερο κύκλο στο δίκτυο που τους περιέχει. Αν όντως υπάρχει τέτοιος κύκλος, να εμφανίζει με τη σειρά τους κόμβους που το αποτελούν, αλλιώς να εμφανίζει κατάλληλο μήνυμα. Στην πρώτη περίπτωση η έξοδος θα πρέπει να είναι της μορφής: $\{v_1, v_2, v_3, v_4, v_5, v_1\}$.

(Προτείνεται για την αναπαράσταση του γράφου να χρησιμοποιήσετε hash map με κλειδί το όνομα του κόμβου (τύπου String) και τιμή τη λίστα των γειτόνων του κόμβου αυτού (τύπου List<String>).)

Άσκηση 3.2

Θεωρήστε το κοινωνικό δίκτυο (social network) του Τμήματός μας, αποτελούμενο από φοιτητές και συνδέσεις μεταξύ τους που αναπαριστούν ερευνητικές συνεργασίες.

Το Τμήμα διοργανώνει μια ημερίδα για την προβολή της ερευνητικής δραστηριότητας σε αυτό. Επειδή δεν υπάρχει χρόνος να παρουσιαστούν τα αποτελέσματα κάθε συνεργασίας ξεχωριστά, οι διοργανωτές σκέφτηκαν να επιλέξουν κάποιους φοιτητές οι οποίοι θα παρουσιάσουν μια σύνοψη των αποτελεσμάτων των συνεργασιών στις οποίες συμμετέχουν. Θέλουν λοιπόν να επιλέξουν το ελάχιστο πλήθος φοιτητών που αρκεί ώστε να παρουσιαστούν όλες οι συνεργασίες.

Έχουν σκεφτεί δύο απλούς αλγόριθμους επίλυσης του προβλήματος αυτού:

1. Εξαντλητική αναζήτηση: Ξεκινάμε με $k = 1$ και ελέγχουμε όλα τα δυνατά υποσύνολα φοιτητών μεγέθους k για να βρούμε κάποιο που αποτελεί λύση του προβλήματός μας. Αν δε βρεθεί λύση, αυξάνουμε το k κατά 1 και επαναλαμβάνουμε.
2. Άπληστη προσέγγιση: Επιλέγουμε το φοιτητή v που έχει το μεγαλύτερο βαθμό στο δίκτυο, τον εισάγουμε στο σύνολο επιλεγμένων φοιτητών, και αφαιρούμε από το δίκτυο όλες τις ακμές που προσπίπτουν στον v . Επαναλαμβάνουμε μέχρι το δίκτυο να μείνει χωρίς ακμές.

Να γράψετε πρόγραμμα Java που παίρνει ως είσοδο ένα δίκτυο σε μορφή λίστας ακμών (παρέχονται test data στην ενότητα 'Εγγραφα' του eClass, στα οποία οι ακμές δίνονται σαν ζεύγη ακεραίων χωρισμένων με tab μεταξύ τους) και επιστρέφει το σύνολο φοιτητών που πρέπει να επιλεγούν έτσι ώστε να παρουσιαστούν όλες οι συνεργασίες καθώς και το χρόνο υπολογισμού του (σε msec), χρησιμοποιώντας κάθε έναν από τους παραπάνω αλγόριθμους. Η έξοδος του προγράμματός σας θα πρέπει να είναι της μορφής:

Exhaustive solution: $\{v_1, v_2, v_3, v_4\}, t_1$

Greedy solution: $\{v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}, t_2$

αν $\{v_1, v_2, v_3, v_4\}$ είναι η λύση που βρίσκει ο 1ος αλγόριθμος και t_1 ο χρόνος υπολογισμού της, και $\{v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$ και t_2 είναι η λύση και ο χρόνος υπολογισμού αντίστοιχα για το 2ο αλγόριθμο.

(Προτείνεται για την αναπαράσταση του γράφου να χρησιμοποιήσετε hash map με κλειδί το όνομα του κόμβου (τύπου String) και τιμή τη λίστα των γειτόνων του κόμβου αυτού (τύπου List<String>).)

Βοήθεια (και για τις δύο ασκήσεις):

Παρέχονται (στην ενότητα 'Εγγραφα' του eClass, στο φάκελο 'Datasets and Test files') ενδεικτικά αρχεία εισόδου που μπορείτε να χρησιμοποιήσετε για να δοκιμάζετε το πρόγραμμά σας. Προφανώς μπορείτε να δημιουργήσετε και να χρησιμοποιήσετε αντίστοιχα δικά σας αρχεία για δοκιμή των υλοποιήσεών σας, προσέχοντας όμως να διατηρήσετε το ίδιο format.

Οδηγίες υποβολής:

- Οι λύσεις των δύο ασκήσεων θα πρέπει να περιέχονται σε διαφορετικά αρχεία (1 class για κάθε άσκηση, πχ. Exercise1.java, Exercise2.java).
- Θα πρέπει να έχετε ένα τρίτο αρχείο για την εξέταση των 2 ασκήσεων (ExercisesSet3.java), με τη μέθοδο 'main'.
- Τα αρχεία που θα χρησιμοποιήσετε για τις δοκιμές σας (πχ 3.1-sm.txt, 3.2-sm.txt), θα πρέπει να βρίσκονται στον ίδιο φάκελο με τα παραπάνω.
- Στην ενότητα 'Εγγραφα', στο φάκελο 'Submission templates' μπορείτε να βρείτε ένα πρότυπο του πώς πρέπει να είναι ο συμπιεσμένος φάκελος που θα υποβάλετε.
- Το pdf αρχείο, καθώς και το αντίστοιχο tex αρχείο για όσους χρησιμοποιήσουν L^AT_EX, τα βάζετε στον ίδιο φάκελο.

Γραπτές Ασκήσεις

Άσκηση 3.3

Να εκτελέσετε το πρόγραμμα που γράψατε στην προγραμματιστική άσκηση 3.2 για δίκτυα διαφόρων μεγεθών, να μετρήσετε το χρόνο εκτέλεσης κάθε αλγορίθμου και να δώσετε σε μια γραφική παράσταση τους χρόνους εκτέλεσης ως συνάρτηση του πλήθους φοιτητών του δικτύου.

Τι τάξης μεγέθους είναι ο χρόνος εκτέλεσης κάθε αλγορίθμου ως προς το πλήθος των φοιτητών και τι παρατηρείτε όσον αφορά τις λύσεις που δίνουν οι δύο αλγόριθμοι; Είναι το ίδιο καλές; Σχετίζονται οι χρόνοι εκτέλεσης με την ποιότητα των λύσεων; Σχολιάστε/δικαιολογήστε σύντομα την απάντησή σας.

Άσκηση 3.4

Θεωρήστε το πρόβλημα του Συνόλου Κυριαρχίας (Dominating Set): Δίνονται ένας μη κατευθυνόμενος γράφος $G = (V, E)$ κι ένας θετικός ακέραιος k και ζητείται να βρεθεί αν υπάρχει σύνολο κυριαρχίας $V' \subseteq V$, δηλαδή σύνολο κόμβων V' τέτοιο ώστε κάθε κόμβος του V είτε να ανήκει στο V' ή να έχει γείτονα στο V' , με μέγεθος k .

Να δείξετε ότι το πρόβλημα αυτό είναι NP-complete.