



ΑΛΓΟΡΙΘΜΟΙ

Διδάσκουσα: Κ. Παπακωνσταντινοπούλου

3η Σειρά Ασκήσεων (Προγραμματιστικές) Προθεσμία υποβολής: 8/6/2020

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. Στον παραπάνω φάκελο θα πρέπει να συμπεριλάβετε και μια αναφορά (pdf αρχείο), που θα περιέχει για κάθε άσκηση μια πολύ σύντομη περιγραφή του αλγορίθμου που χρησιμοποιήσατε και την πολυπλοκότητά του, καθώς και τις απαντήσεις στα ερωτήματα της άσκησης 3.5.

Άσκηση 3.1

Να γράψετε πρόγραμμα Java που υλοποιεί τον διαίρει-και-βασίλευε αλγόριθμο που σχεδιάσατε στην άσκηση 1.4 της 1ης σειράς ασκήσεων.

Άσκηση 3.2

Να υλοποιήσετε ένα κωδικοποιητή Huffman τριών συμβόλων (0, 1 και 2) που δέχεται στην είσοδο ένα κείμενο γραμμένο στην αγγλική γλώσσα και επιστρέφει τη βέλτιστη κωδικοποίησή του, δηλαδή την κωδικοποίηση που έχει το μικρότερο δυνατό μέγεθος.

Άσκηση 3.3

Να γράψετε πρόγραμμα Java που υλοποιεί τον αλγόριθμο δυναμικού προγραμματισμού που σχεδιάσατε στην άσκηση 1.6 της 1ης σειράς ασκήσεων.

Άσκηση 3.4

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/philosophyedgelist.txt> και μπορείτε να το ανοίξετε αφού κάνετε login στο eClass.

Να γράψετε πρόγραμμα Java που παίρνει ως είσοδο το δίκτυο αυτό, υπολογίζει τη διάμετρό του, και αν αυτή ορίζεται από το μονοπάτι από τον κόμβο x στον y , να εμφανίζει τους κόμβους του μονοπατιού με τη σειρά που εμφανίζονται σε αυτό. Η έξοδος θα πρέπει να είναι της μορφής: $\{x, v_1, v_2, v_3, v_4, v_5, y\}$.

(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 3.5

Θεωρήστε ένα κοινωνικό δίκτυο (social network) του Πανεπιστημίου μας, αποτελούμενο από προσωπικό και φοιτητές, και συνδέσεις μεταξύ τους που αναπαριστούν ερευνητικές συνεργασίες. Θεωρούμε ότι κάθε μέλος του προσωπικού και κάθε φοιτητής μπορεί να συμμετέχει σε όσες συνεργασίες επιθυμεί.

Θέλουμε να δεσμεύσουμε μια αίθουσα αποκλειστικά για συναντήσεις των ερευνητικών ομάδων. Δεδομένου ότι η αίθουσα αυτή δε θα είναι πλέον διαθέσιμη για μαθήματα, προτιμούμε να δεσμεύσουμε μια μικρή αίθουσα, που όμως θα πρέπει να χωράει ακόμα και την πιο μεγάλη ομάδα. Θέλουμε λοιπόν να εντοπίσουμε το μέγεθος της πιο πολυπληθούς συνεργασίας που βρίσκεται σε εξέλιξη, δηλαδή το μεγαλύτερο υποσύνολο ατόμων του παραπάνω δικτύου στο οποίο κάθε μέλος του συνεργάζεται με όλα τα υπόλοιπα (μέγιστη κλίκα).

Έχουμε σκεφτεί δύο απλούς αλγόριθμους επίλυσης του προβλήματος αυτού:

- α. 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 βρεθεί λύση, αυξάνουμε το k κατά 1 και επαναλαμβάνουμε. Ο αλγόριθμος επιστρέφει τη λύση που βρήκε για το μέγιστο k στο οποίο βρέθηκε λύση.
- β. Άπληστη προσέγγιση: Θεωρούμε μια αυθαίρετη διάταξη (v_1, v_2, \dots, v_n) των κόμβων του γράφου και αρχικοποιούμε τη ζητούμενη κλίκα σε $\{v_1\}$. Εξετάζουμε τους υπόλοιπους κόμβους με την παραπάνω σειρά, και όποιον συνδέεται με όλους τους κόμβους της κλίκας, τον εισάγουμε στην κλίκα. Ο αλγόριθμος επιστρέφει την κλίκα που βρήκε.

Να γράψετε πρόγραμμα Java που παίρνει ως είσοδο ένα δίκτυο σε μορφή λίστας ακμών (παρέχονται test data στην ενότητα 'Έγγραφα' του eClass, στα οποία οι ακμές δίνονται σαν ζεύγη ακεραίων χωρισμένων με tab μεταξύ τους) και επιστρέφει την πιο πολυπληθή ερευνητική ομάδα καθώς και το χρόνο υπολογισμού της (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: $\{v_6, v_7, v_8, v_{10}\}, t_2$

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 $\{v_6, v_7, v_8, v_{10}\}$ 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>)).

Να εκτελέσετε το πρόγραμμα που γράψατε 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.

Βοήθεια:

Παρέχονται (στην ενότητα 'Έγγραφα' του eClass, στο φάκελο 'Auxiliary source files') συναρτήσεις για το διάβασμα ενός αρχείου που περιέχει ακεραίους χωρισμένους με ένα κενό μεταξύ τους ή διδιάστατο πίνακα ακεραίων χωρισμένων με ένα κενό μεταξύ τους και την αποθήκευσή τους σε μια λίστα ή ένα πίνακα αντίστοιχα.

Παρέχονται (στην ενότητα 'Έγγραφα' του eClass, στο φάκελο 'Datasets and Test files') ενδεικτικά αρχεία εισόδου που μπορείτε να χρησιμοποιήσετε για να δοκιμάζετε τα προγράμματά 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 ασκήσεων θα πρέπει να περιέχονται σε διαφορετικά αρχεία (1 class για κάθε άσκηση, πχ. Exercise1.java, Exercise2.java, κ.ο.κ.).
- Θα πρέπει να έχετε ένα ξεχωριστό αρχείο για την εξέταση των ασκήσεων (ExerciseSet3.java), με τη μέθοδο 'main'.
- Οι είσοδοι για όλα τα προγράμματα που θα γράψετε θα πρέπει να διαβάζονται από αρχείο.
- 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 'Έγγραφα', 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.