

## 4η Σειρά Προγραμματιστικών Ασκήσεων

### Άσκηση

1. Σχεδιάστε και υλοποιήστε αλγόριθμο εύρεσης συνεκτικών συνιστωσών σε μη κατευθυνόμενο γράφο με κατάλληλη τροποποίηση της διαδικασίας  $dft(G)$  (ίδια με την  $bft(G)$  των διαφανειών, όπου αντί για  $bfs$  χρησιμοποιείται  $dfs$ ). Θα πρέπει σε κάθε κόμβο να αποδίδεται ένας αριθμός, που να δηλώνει την συνιστώσα στην οποία ανήκει.
2. Σχεδιάστε και υλοποιήστε αλγόριθμο εντοπισμού κύκλων σε μη κατευθυνόμενο γράφο με κατάλληλη τροποποίηση της διαδικασίας  $dft$ .
3. Σχεδιάστε και υλοποιήστε αλγόριθμο παρόμοιο με την  $dft(G)$  για διάσχιση κατευθυνόμενων γράφων, που να επιστρέφει επιπλέον την πληροφορία ύπαρξης ή μη κατευθυνόμενων κύκλων στον γράφο.  
*Υπόδειξη: χρησιμοποιήστε τις μη δενδρικές ακμές, που κατευθύνονται από έναν κόμβο σε πρόγονό του, γνωστές ως “back edges” ή “ανιούσες ακμές”. Αυτές μπορούν να εντοπιστούν με χρήση δύο δεικτών για κάθε κόμβο, που εκφράζουν το βήμα πρώτης και τελευταίας επίσκεψης του κόμβου. Για λεπτομέρειες δείτε βιβλία Cormen-Leiserson-Rivest-Stein και Dasgupta-Papadimitriou-Vazirani.*
4. Χρησιμοποιήστε την αρίθμηση με δύο δείκτες του προηγούμενου ερωτήματος προκειμένου να υλοποιήσετε αλγόριθμο που επιτυγχάνει “τοπολογική ταξινόμηση” (topological sorting) ενός κατευθυνόμενου ακυκλικού γράφου (directed acyclic graph - DAG), δηλαδή αρίθμηση όπου κάθε ακμή πηγαίνει από μικρότερο σε μεγαλύτερο κόμβο.
5. Σχεδιάστε και υλοποιήστε αλγόριθμο εύρεσης ισχυρά συνεκτικών συνιστωσών σε κατευθυνόμενο γράφο με χρήση δύο κλήσεων διαδικασιών τύπου  $dft(G)$  παρόμοιων με αυτές των ερωτημάτων 4 και 5.

*Υπόδειξη: μια από τις κλήσεις του DFS θα πρέπει να γίνει στον “αντεστραμμένο γράφο”, δηλαδή στον γράφο όπου κάθε ακμή έχει αντιστραφεί, και να χρησιμοποιηθεί ο δείκτης τελευταίας επίσκεψης. Για περισσότερες λεπτομέρειες ανατρέξτε στα βιβλία που αναφέρονται παραπάνω.*

**Να παραδοθούν μέχρι την εβδομάδα 1-5/12/2008**