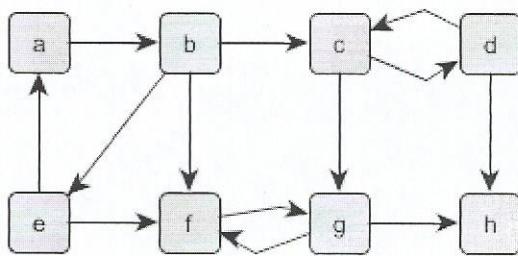


Αλγόριθμοι και Πολυπλοκότητα
4 Σεπτεμβρίου 2017

- Διάρκεια: 2 ½ ώρες.
- Καλή επιτυχία.

Θέμα 1^ο

Να υπολογίσετε τις ισχυρά συνδεδεμένες συνιστώσες για το παρακάτω κατευθυνόμενο γράφημα. Να παρουσιάσετε την δουλειά σας λεπτομερειακά.

**Θέμα 2^ο**

Έστω ένα κατευθυνόμενο, με βάρη, ακυκλικό γράφημα $G=(V,E)$ και μια συνάρτηση βάρους $w:E \rightarrow N^+$. Μία τοπολογική αρίθμηση με βάρη, $\text{number}()$, του G είναι μία ανάθεση αριθμών στις κορυφές του G τέτοια ώστε για κάθε ακμή $(u,v) \in E$ ισχύει $\text{number}(v) \geq \text{number}(u) + w((u,v))$. Να σχεδιάσετε έναν αλγόριθμο ο οποίος υπολογίζει μία τοπολογική αρίθμηση με βάρη για το γράφημα G .

Να αιτιολογήσετε την ορθότητα του αλγορίθμου σας. Να αναλύσετε την πολυπλοκότητά του.

Θέμα 3^ο

Έστω διάνυσμα A με n στοιχεία. Ένα στοιχείο του A καλείται πλειοψηφικό εάν εμφανίζεται περισσότερες από $n/2$ φορές. **Σχεδιάστε** και **αναλύστε** έναν αλγόριθμο τύπου "διαίρει και βασίλευε" ο οποίος θα εντοπίζει το πλειοψηφικό στοιχείο (εάν υπάρχει) σε χρόνο $n \log n$. Δεν επιτρέπεται να μεταχειριστείτε αλγόριθμο ταξινόμησης, καθώς θεωρούμε ότι διαθέσιμη είναι μόνο η πράξη του ελέγχου ισότητας μεταξύ στοιχείων του διανύσματος. **Να επιχειρηματολογήσετε** ως προς την ορθότητα του αλγορίθμου σας.

Θέμα 4^ο

Μία γλώσσα επεξεργασίας αλφαριθμητικών (strings) περιλαμβάνει την λειτουργία της «διάσπασης» ενός αλφαριθμητικού σε 2 νέα τμήματα. Επειδή η λειτουργία εμπεριέχει την αντιγραφή του αρχικού αλφαριθμητικού, η διάσπαση ενός αλφαριθμητικού n χαρακτήρων σε δύο νέα αλφαριθμητικά κοστίζει n μονάδες χρόνου. Έστω ότι ένας προγραμματιστής θέλει να σπάσει ένα αλφαριθμητικό σε πολλά τμήματα. Η σειρά με την οποία γίνονται τα σπασίματα μπορεί να επηρεάσει το συνολικό απαιτούμενο χρόνο. Για παράδειγμα, έστω ότι θέλουμε να σπάσουμε ένα αλφαριθμητικό 20 χαρακτήρων μετά τους χαρακτήρες στις θέσεις 3, 8 και 10, δημιουργώντας έτσι 4 νέα αλφαριθμητικά. Εάν η διάσπαση γίνει με την από «*αριστερά προς τα δεξιά*» σειρά των θέσεων διάσπασης, τότε θα χρειαστούν 20 μονάδες χρόνου για την πρώτη διάσπαση, 17 μονάδες χρόνου για την δεύτερη και 12 μονάδες χρόνου για την τρίτη, δηλαδή συνολικά 49 μονάδες χρόνου. Εάν η διάσπαση γίνει με την από «*δεξιά προς τα αριστερά*» σειρά των θέσεων διάσπασης, τότε θα χρειαστούν 20 μονάδες χρόνου για την πρώτη διάσπαση, 10 μονάδες χρόνου για την δεύτερη και 8 μονάδες χρόνου για την τρίτη, δηλαδή συνολικά 38 μονάδες χρόνου.

Σχεδιάστε έναν αλγόριθμο δυναμικού προγραμματισμού ο οποίος, με δεδομένη μία λίστα από τις επιθυμητές θέσεις διάσπασης ενός αλφαριθμητικού n χαρακτήρων, υπολογίζει την βέλτιστη σειρά διάσπασης σε $O(n^2)$ μονάδες χρόνου.