Αλγόριθμοι και Πολυπλοκότητα

Σετ Ασκήσεων 2 – Δυναμικός Προγραμματισμός, Γραφήματα

(Παράδοση: έως ΠΕΜΠΤΗ 13/1/2022, 12:00)

Απαντήστε όλες τις παρακάτω ερωτήσεις σε χαρτί. Μπορείτε να τις δακτυλογραφήσετε αν θέλετε, αλλά δεν είναι απαραίτητο και μπορεί να ταλαιπωρηθείτε. Οι απαντήσεις θα πρέπει να είναι ΚΑΘΑΡΟΓΡΑΜΜΕΝΕΣ, διότι θα αφαιρεθούν βαθμοί για ασαφείς και ακατάληπτες απαντήσεις. Μπορείτε να θεωρήσετε ως δεδομένα οποιονδήποτε αλγόριθμο και αποτέλεσμα έχετε διδαχθεί στο μάθημα. Θα πρέπει να παραδώσετε τις απαντήσεις σας στους βοηθούς του μαθήματος.

ΠΡΟΣΟΧΗ. Θα υπάρξει προφορική εξέταση της άσκησης από τους βοηθούς του μαθήματος ώστε να διαπιστώσουν κατά πόσον έχετε πραγματικά δουλέψει την άσκηση και καταλαβαίνετε τις λύσεις που παραδίδετε. Η προφορική αυτή εξέταση θα μετρήσει για το 30% του βαθμού της άσκησης (δηλ., όποιος δεν θέλει να εξεταστεί θα βαθμολογηθεί με άριστα το 70/100).

Η αντιμετώπισή φαινομένων αντιγραφής θα είναι αυστηρή.

Ασκήσεις

1. (30 ΜΟΝΑΔΕΣ) Έστω μία μήκους-n ακολουθία αριθμών $f = f_1, f_2, \ldots, f_n$ – για παράδειγμα, σε μία βαση δεδομένων με μετρήσεις θερμοκρασίας, το f_i μπορεί να αναπαριστά την μέτρηση για την χρονική στιγμή i. Ένα βασικό πρόβλημα στις βάσεις δεδομένων είναι να βρεθεί μία προσέγγιση της ακολουθίας f η οποία να χρησιμοποιεί μόνο B αριθμούς (όπου $B << n$). Αυτό τυπικά γίνεται με ένα ιστόγραμμα, το οποίο διαμερίζει το $[1, 2, \ldots n]$ σε B συνεχόμενα διαστήματα (buckets) $b_i = [s_i, s_i + 1, \ldots, e_i]$ (όπου $s_{i+1} = e_i + 1$) για $i = 1, \ldots, B$, και χρησιμοποιεί μόνο έναν αριθμό (π.χ., την μέση τιμή) σαν προσέγγιση για όλες τις τιμές σε ένα bucket.

   Αυτό βέβαια συνεπάγεται ένα προσεγγιστικο λάθος για κάθε bucket $b_i = [s_i, \ldots, e_i]$, έστω $ERR(s_i, e_i)$. Ο στόχος μας είναι να βρούμε ένα ιστόγραμμα (δηλ., μία διαμέριση του $[1, \ldots, n]$) το οποίο να ελαχιστοποιεί το συνολικό προσεγγιστικό λάθος, δηλ., να ελαχιστοποιεί την ποσότητα $\sum_{i=1}^{B} ERR(s_i, e_i)$.

   Περιγράψτε έναν (πολυωνυμικό) αλγόριθμο ο οποίος θα δέχεται σαν είσοδο τα n, B, και την ακολουθία $f = f_1, \ldots, f_n$, και θα επιστρέφει το βέλτιστο (ελαχίστου συνολικού λάθος) ιστόγραμμα με B buckets για την f. Υποθέστε ότι υπάρχει μία κόστους-$O(1)$ υπο-ρουτίνα $ERR(s, e)$ που επιστρέφει το προσεγγιστικό λάθος για ένα δοθέν bucket $[s, s + 1, \ldots, e]$. Ποια είνα η πολυπλοκότητα του αλγορίθμου σας;

   [[Υπόδειξη: Σκεφτείτε αναδρομικά με βάση ΔΠ: Έστω $E[i, b]$ το βέλτιστο (ελάχιστο) λάθος για την υπακολουθία $f_1, f_2, \ldots, f_i$ κάνοντας χρήση b buckets. Πως μπορεί να εκφραστεί το $E[i, b]$

σαν συνάρτηση μικροτέρων προβλημάτων; Η βασική επιλογή για την αναδρομή σας θα είναι το αριστερό όριο j του τελευταίου **bucket** στην λύση $E[i, b]$ (δηλ. καλύπτει την υπακολουθία $f_j, f_{j+1}, \ldots, f_i$).]]

2. (25 ΜΟΝΑΔΕΣ) Δίνεται ένα σύνολο από n θετικούς ακεραίους $\{l_1, \ldots, l_n\}$ και ένας θετικός ακέραιος B. Το ζητούμενο είναι να σχεδιάσετε έναν αλγόριθμο ο οποίος θα επιστρέφει **TRUE** αν και μόνο αν υπάρχει κάποιο υποσύνολο $S \subseteq \{l_1, \ldots, l_n\}$ τέτοιο ώστε $\sum_{l_i \in S} = B$. (α) Ποια είναι η πολυπλοκότητα μιας εξαντλητικής (**brute force**) λύσης για το πρόβλημα; (β) Γράψτε μία αναδρομή Δυναμικού Προγραμματισμού (ΔΠ) η οποία να λύνει το πρόβλημα. Ποιά είναι η πολυπλοκότητα του αντίστοιχου αλγορίθμου ΔΠ; Είναι πολυωνυμική;

3. (25 ΜΟΝΑΔΕΣ) Έχουμε ένα σύνολο από n εργασίες. Η εργασία i απαιτεί χρόνο t_i, $1 \leq i \leq n$. Οι εργασίες μπορούν να εκτελούνται παράλληλα, αλλά κάθε εργασία έχει προαπαιτούμενες εργασίες, που πρέπει να περιμένει να ολοληρωθούν πριν αρχίσει. Έστω $S_i \subseteq \{1, \ldots, n\}$ οι προαπαιτούμενες εργασίες για την εργασία i. Η εργασία i μπορεί να ξεκινήσει αμέσως αφού ολοκληρωθούν όλες οι προαπαιτούμενες εργασίες στο S_i.

   Σχεδιάστε (σε ψευδοκώδικα, ή, αν προτιμάτε, σε **C**, **C++** ή **Java**), εξηγήστε, και δώστε την ανάλυση για έναν αλγόριθμο γραμμικού χρόνου $O(n + m)$ (όπου $m = \sum_i |S_i|$) ο οποίος θα υπολογίζει τον ελάχιστο δυνατό χρόνο ολοκλήρωσης για κάθε εργασία.

4. (20 ΜΟΝΑΔΕΣ) Ο ορισμός του Ελαφρύτατου Συνδετικου Δένδρου (ΕΣΔ) που περιγράψαμε θεωρεί ως βάρος $w(T)$ ενός δένδρου το άθροισμα των βαρών των ακμών του, δηλ. $w(T) = \sum_{e \in T} w(e)$.

   (α′) (10 ΜΟΝΑΔΕΣ) Αποδείξτε ότι οι αλγόριθμοι κατασκευής ΕΣΔ που μελετήσαμε (**Prim**, **Kruskal**) μπορούν να εφαρμοστούν και στην περίπτωση που ως βάρος $w(T)$ ενός δέντρου T ορίζουμε το μέγιστο από τα βάρη των ακμών του, δηλ. $w(T) = \max_{e \in T}\{w(e)\}$.

   (β′) (10 ΜΟΝΑΔΕΣ) Ισχύει το ίδιο εάν ορίσουμε το βάρος του δένδρου σαν το γινόμενο των βαρών, δηλ. $w(T) = \Pi_{e \in T} w(e)$; Αν δεν ισχύει γενικά, υπάρχουν κάποιες επιπλέον συνθήκες κάτω από τις οποίες να ισχύει;