

ΕΠΑ 232: Αλγόριθμοι και Πολυπλοκότητα

Κατ'οίκον Εργασία 1

Ημερομηνία Παράδοσης: 26/02/02

1. Να προτείνετε διαίρει και βασίλευε αλγόριθμο ο οποίος, με δεδομένο εισόδου ένα πίνακα από n ακέραιους, βρίσκει και επιστρέφει το ελάχιστο και το μέγιστο στοιχείο του πίνακα σε $\lceil 3n/2 \rceil - 2$ συγκρίσεις.
2. Έχετε να τοποθετήσετε n βιβλία b_1, \dots, b_n , στα ράφια μιας βιβλιοθήκης με τη σειρά που σας δίνονται. Για κάθε $1 \leq i \leq n$ το βιβλίο b_i έχει πάχος t_i και ύψος h_i . Το μήκος κάθε ραφιού της βιβλιοθήκης είναι L .
 - (α) Υποθέστε πως τα ράφια έχουν προκαθορισμένο ύψος h και $h_i < h$ για κάθε i . Να δώσετε άπληστο αλγόριθμο ο οποίος αποφασίζει τοποθέτηση των βιβλίων που χρησιμοποιεί τον ελάχιστο δυνατό αριθμό ραφιών. Ποια η πολυπλοκότητα του αλγόριθμού σας; Να αιτιολογήσετε την ορθότητά του με αυστηρά επιχειρήματα.
 - (β) Τώρα υποθέστε πως τα ράφια έχουν μεταβλητό ύψος, δηλαδή μπορείτε να αποφασίσετε εσείς το ύψος του καθενός από αυτά ανάλογα με το πιο ψηλό βιβλίο που περιέχει. Θέλετε να βρείτε τοποθέτηση βιβλίων που ελαχιστοποιεί το συνολικό ύψος των ραφιών που χρησιμοποιούνται.
 - (i) Να δείξετε πως ο άπληστος αλγόριθμος που προτείνετε στο μέρος (α) δεν δίνει πάντα βέλτιστη λύση στο πρόβλημα.
 - (ii) Έστω $H[m]$, το ελάχιστο συνολικό ύψος που απαιτείται για την τοποθέτηση των βιβλίων b_m, \dots, b_n , σε ράφια. Να ορίσετε αναδρομικά την τιμή του $H[m]$. (Βεβαιωθείτε πως η απάντησή σας καλύπτει όλες τις περιπτώσεις.) Με βάση την πιο πάνω σχέση να σχεδιάσετε αλγόριθμο δυναμικού προγραμματισμού ο οποίος να λύνει το πρόβλημα. Ποιος ο χρόνος εκτέλεσης του αλγόριθμού σας;
 - (iii) Να επεκτείνετε τον αλγόριθμό σας έτσι ώστε να υπολογίζεται όχι μόνο το συνολικό ύψος της βέλτιστης τοποθέτησης αλλά και στοιχεία για τη συγκεκριμένη τοποθέτηση (δηλαδή ποια βιβλία τοποθετούνται σε κάθε ράφι). Ποια η χρονική πολυπλοκότητα του αλγόριθμού σας σαν συνάρτηση του n ;
3. Πρόβλημα πολλαπλασιασμού ακεραίων
 - (α) Έχει προταθεί αλγόριθμος οποίος πετυχαίνει πολλαπλασιασμό δύο ακεραίων με n bits ο καθένας, με 6 αναδρομικούς πολλαπλασιασμούς και 6 προσθέσεις ακεραίων με $n/3$ bits. Να γράψετε αναδρομική εξίσωση η οποία να εκφράζει το χρόνο εκτέλεσης του αλγόριθμου και να τη λύσετε με τη μέθοδο της επανάληψης.

(β) Προφανώς ο αλγόριθμος που αναφέρεται στο μέρος (α) δεν βελτιώνει τον αλγόριθμο που μελετήσαμε στο Φροντιστήριο 1. Έστω ότι γνωρίζουμε την ύπαρξη ενός αλγόριθμου A ο οποίος πολλαπλασιάζει δύο ακέραιους με n bits ο καθένας, με m αναδρομικούς πολλαπλασιασμούς και κάποιο σταθερό αριθμό προσθέσεων ακεραίων με $n/3$ bits. Να βρείτε τη μέγιστη τιμή του m για την οποία ο αλγόριθμος A είναι αποδοτικότερος από τον αλγόριθμο πολλαπλασιασμού ακεραίων που μελετήσαμε στο Φροντιστήριο 1. (Μπορείτε να χρησιμοποιήσετε το θεώρημα γενικής χρήσης για λύση οποιασδήποτε αναδρομικής εξίσωσης συναντήσετε.

4. Θεωρήστε μια πόλη οι δρόμοι της οποίας ορίζουν μια σχάρα $X \times Y$. Θέλουμε να μεταβούμε από την αριστερότερη γωνιά στο πάνω μέρος της σχάρας στη δεξιότερη γωνιά στο κάτω μέρος της σχάρας. Στοιχείο που δυσχεραίνει την αποστολή μας είναι ότι η πόλη περιέχει κάποιες κακόφημες περιοχές τις οποίες θα πρέπει να αποφύγουμε. Έστω ότι οι περιοχές αυτές μας δίνονται μέσω ενός δισδιάστατου πίνακα $BAD[X, Y]$, όπου $BAD[x, y] = \text{true}$ αν και μόνο αν η διασταύρωση των δρόμων x και y είναι ένα από τα σημεία που πρέπει να αποφύγουμε.

(i) Να δώσετε ένα στιγμιότυπο του πίνακα BAD , για το οποίο δεν υπάρχει κανένα μονοπάτι ανάμεσα στο σημείο εκκίνησης και το σημείο προορισμού μας το οποίο να αποφεύγει τις κακόφημες περιοχές.

(ii) Να γράψετε ένα αλγόριθμο οπισθοδρόμησης ο οποίος, με δεδομένο εισόδου τον πίνακα BAD , να βρίσκει κάποιο μονοπάτι που ικανοποιεί τις προδιαγραφές του προβλήματος, αν υπάρχει, διαφορετικά να αναφέρει την απουσία ενός τέτοιου μονοπατιού. Να σχολιάσετε την ορθότητα του αλγόριθμού σας και το χρόνο εκτέλεσής του.

(iii) Να δώσετε αλγόριθμο δυναμικού προγραμματισμού ο οποίος, με δεδομένο εισόδου τον πίνακα BAD , υπολογίζει και επιστρέφει το συντομότερο μονοπάτι που ικανοποιεί τις προδιαγραφές του προβλήματος. Να επιδιώξετε ο αλγόριθμός σας να έχει χρόνο εκτέλεσης $O(X \cdot Y)$