Πανεπιστημιο Κρητης

Τμήμα Εφαρμοσμένων Μαθηματικών

Εξέταση Σεπτεμβρίου 2008

Θωρία Αλγορίθμων (ΕΜ 202)

ΘΕΜΑΤΑ

1A) (μονάδες 3): Περιγράψτε τη διαφορά μεταξύ της ανάλυσης αναμενόμενης και χειρότερης περίπτωσης για έναν αιτιοκρατικό αλγόριθμο. Δώστε παράδειγμα αιτιοκρατικού αλγορίθμου του οποίου οι δύο περιπτώσεις διαφέρουν.

Λύση: Στην ανάλυση αναμενόμενης περίπτωσης για μία κατανομή εισόδου του αλγορίθμου υπολογίζουμε τον αναμενόμενο χρόνο εκτέλεσης ως προς τη συγκεκριμένη κατανομή. Στην ανάλυση χειρότερης περίπτωσης εστιάζουμε σε εκείνες τις εισόδους που αναγκάζουν τον αλγόριθμο να "τρέχει" περισσότερο χρόνο. Παράδειγμα είναι η $QuickSort$ που έχει χειρότερη περίπτωση $\Theta(n^2)$, όταν η συστοιχία εισόδου είναι ήδη ταξινομημένη, αλλά έχει αναμενόμενο χρονο $\Theta(n\lg n)$.

1B) (μονάδες 6): Λύστε τις πιο κάτω αναδρομικές εξισώσεις με το βασικό θεώρημα:

1) $T(n) = T(2n/3) + 1$,

2) $T(n) = 3T(n/4) + n\log n$.

Λύση:

1) $a = 1, b = 3/2, f(n) = 1$. Άρα $f(n) = \Theta(n^{log_b a}) = \Theta(1)$ άρα περίπτωση 2, άρα $T(n) = \Theta(\lg n)$.

2) $a = 3, b = 4, f(n) = n\lg n$. Επίσης $n^{log_b a} = n^{log_4 3} = O(n^{0.793})$. Άρα $f(n) = \Omega(n^{log_4 3} + \varepsilon)$, $\varepsilon \approx 0.2$. Επιπλέον ισχύει και η συνθήκη ομαλότητας, διότι $af(\frac{n}{b}) = 3(\frac{n}{4})\log(\frac{n}{4}) \leq 3\frac{n}{4}\log n = cf(n)$, για $c = \frac{3}{4}$. Άρα περίπτωση 3 και $T(n) = \Theta(n\log n)$.

2A) (μονάδες 4): Ποιά τα βασικά χαρακτηριστικά που θα πρέπει να διαθέτει ένα πρόγραμμα βελτιστοποίησης ώστε να μπορεί να εφαρμοστεί η μέθοδος του δυναμικού προγραμματισμού; Ποιά τα βασικά χαρακτηριστικά που θα πρέπει να διαθέτει ένα πρόγραμμα βελτιστοποίησης ώστε να μπορεί να εφαρμοστεί η άπληστη στρατηγική;

Λύση: Βιβλίο σελ. 339 και σελ. 382.

2B) (μονάδες 5): Κάποια υλοποίηση του αλγορίθμου $Quicksort$ πάντοτε επιλέγει να κάνει τη διαμέριση ως προς το στοιχείο $A[3]$, όπου $A[1 \ldots n]$ είναι ο πίνακας που διαμερίζουμε. Ποιά είναι η ανάλυση χειρότερης περίπτωσης για αυτή την υλοποίηση;

Λύση: Η είσοδος που κάνει την πιο πάνω υλοποίηση να τρέχει στο χειρότερο χρόνο ειναι ο πίνακας $A[n, n-1, 1, 2, 3, \ldots, n-2]$, για την οποία παίρνουμε $T(n) = T(n-2) + \Theta(n)$, όπου $\Theta(n)$

το κόστος της διαμέρισης και $T(n-2)$ το κόστος του υποπροβλήματος που πρέπει να λύσουμε μετά τη διαμέριση. Η πιο πάνω έχει λύση $T(n) = \Theta(n^2)$.

2Γ) (μονάδες 10): Θεωρείστε σκακιέρα (πίνακα) $n \times n$ και μία συνάρτηση κόστους $c(i,j)$ για κάθε τετράγωνο του πίνακα (i οι γραμμές, j οι στήλες). Υποθέστε οτι κάποιος ξεκινά από τυχαίο τετράγωνο της πρώτης γραμμής και μπορεί να κινηθεί κατά ένα τετράγωνο είτε μπροστά, είτε λοξά δεξιά, είτε λοξά αριστερά (λοξά = διαγώνια) (π.χ. αν βρισκόμαστε στο κελί $(2,2)$ τότε οι επιτρεπτές κινήσεις είναι να πάμε είτε στο $(3,2)$, είτε στο $(3,1)$, είτε στο $(3,3)$). Αν κάθε φορά το κόστος του τετραγώνου που επισκέπτομαι προστίθεται στο συνολικό κόστος που πρέπει να πληρώσω, ποιό είναι το ελάχιστο κόστος για να φτάσω στην τελευταία γραμμή του πίνακα;

Λύση: Έστω $q(i,j)$ το ελάχιστο (minimum) κόστος για να φθάσω το (i,j). Προφανώς το πρόβλημα παρουσιάζει την ιδιότητα της βέλτιστης υποδομής. Δηλ. η λύση ολόκληρου του προβλήματος εξαρτάται από τις λύσεις των υποπροβλημάτων. Αν $(i,j) = A$ τότε στο A μπορώ να φθάσω είτε από το $(i-1,j-1)$ είτε από το $(i-1,j)$ είτε από το $(i-1,j+1)$.

$$\text{Άρα } (i,j) = \begin{cases} \infty & ,j < 1 \text{ ή } j > n \\ c(i,j) & ,i = 1 \\ min(q(i-1,j-1), q(i-1,j), q(i-1,j+1)) + c(i,j) & \text{, αλλιώς.} \end{cases}$$

3Α) (μονάδες 5): Υπάρχει συγκριτικός αλγόριθμος ταξινόμησης που να ταξινομεί πίνακα 6 στοιχείων μόνο με 9 συγκρίσεις; Αιτιολογείστε.

Λύση: Από το δένρο απόφασης βλέπουμε ότι για να ταξινομήσουμε έναν πίνακα 6 στοιχείων χρειαζόμαστε τουλάχιστον 6!=720 φύλλα. Όμως με 9 συγκρίσεις μπορούμε να πάρουμε δένδρο μόνο με $2^9 = 512 < 720$ φύλλα. Άρα δεν υπάρχει τέτοιος αλγόριθμος.

3Β) (μονάδες 3): Έστω πίνακας $A[1..n]$. Μπορεί να βρεθεί το στοιχείο με τις πιο πολλές εμφανίσεις στον A σε χρόνο $O(n\lg n)$; Εξηγείστε.

Λύση: Ναι μπορεί ως εξής: Ταξινομώ τον A σε χρόνο $\Theta(n\lg n)$ (χρησιμοποιώ για παράδειγμα συγχωνευτική ταξινόμηση), και έπειτα διατρέχω τον ταξινομημένο A (σε χρόνο $\Theta(n)$) για να βρώ τη μακρύτερη ακολουθία από ίδια στοιχεία. Ο πιο πάνω αλγόριθμος είναι $\Theta(n\lg n)$ που είναι και $O(n\lg n)$.

3Γ) (μονάδες 14): Σε μία δομή δεδομένων εκτελείται μία ακολουθία n πράξεων. Η i-οστή πράξη κοστίζει i εάν το i είναι κάποια δύναμη του 2, και 1 διαφορετικά. Προσδιορίστε το αντισταθμιστικό κόστος ανά πράξη μέσω της χρεωπιστωτικής μεθόδου ανάλυσης.

Λύση: Έστω c_i το κόστος της i-οστής πράξης. Θα ισχύει οτι $c_i = \begin{cases} i & \text{αν } i \text{ δύναμη του 2,} \\ 1 & \text{αλλιώς.} \end{cases}$

Πιστώνουμε σε κάθε πράξη 3 euro (αυτό είναι το αντισταθμιστικό (λογιστικό) κόστος $\hat{c}_i$). Αν i δεν είναι ακριβή δύναμη του 2, πληρώνουμε 1 euro, και αποθηκεύουμε 2 euro. Αν i είναι ακριβή δύναμη του 2, πληρώνουμε i από αυτά που έχουμε αποθηκεύσει. Ο πίνακας που μου δείχνει τα διάφορα κόστη είναι ο πιο κάτω:

Πράξη	Κόστος	Πραγματικό κόστος	Πίστωση
1	3	1	2
2	3	2	3
3	3	1	5
4	3	4	4
5	3	1	6
6	3	1	8
7	3	1	10
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Ισχύει ότι: $\sum_{i=1}^{n} \hat{c}_i = 3n \leq n + \sum_{j=0}^{lgn} 2^j = n - (2n-1) < 3n$. Άρα $\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i \Rightarrow$ πίστωση = λογιστικό - πραγματικό κόστος ≥ 0. Αφού λοιπόν το λογιστικό κόστος κάθε πράξης είναι $O(1)$, και η πίστωση ποτέ δεν γίνεται αρνητική, το συνολικό κόστος των n πράξεων είναι $O(n)$.