

Άσκηση 1

Ημερομηνία Παράδοσης: 11:00, 24 Μαρτίου 2010

Πρόβλημα 1 [15 μονάδες] Χρησιμοποιήστε το κεντρικό θεώρημα για να προσδιορίσετε αυστηρά ασυμπτωτικά φράγματα για τις ακόλουθες αναδρομικές σχέσεις:

1. $T(n) = 4T(n/2) + n$.
2. $T(n) = 4T(n/2) + n^2$.
3. $T(n) = 4T(n/2) + n^3$.

Πρόβλημα 2 [20 μονάδες] Μπορεί να εφαρμοστεί στην αναδρομική σχέση

$$T(n) = 4T(n/2) + n^2 \lg n$$

η κεντρική μέθοδος; Αιτιολογήστε την απάντησή σας. Προσδιορίστε ένα ασυμπτωτικό άνω φράγμα για τη σχέση αυτή.

Πρόβλημα 3 [10 μονάδες] Με χρήση των ορισμών δείξτε ότι: $f(n) + o(f(n)) = \Theta(f(n))$.

Πρόβλημα 4 [15 μονάδες] Θεωρήστε την αναδρομική σχέση $T(n) = T(\lg n) + 1$. Δείξτε ότι $T(n) = O(\lg^* n)$.

Πρόβλημα 5 [40 μονάδες] Μία ακολουθία αντικειμένων a_1, a_2, \dots, a_n θα ονομάζεται *κυρτή* αν υπάρχει k , με $1 \leq k \leq n$, τέτοιο ώστε για κάθε $1 \leq i \leq k - 1$ να έχουμε $a_i \geq a_{i+1}$, ενώ για κάθε $k \leq i \leq n - 1$ να έχουμε $a_i \leq a_{i+1}$.

Σχεδιάστε έναν αλγόριθμο γραμμικού χρόνου ο οποίος ταξινομεί τα στοιχεία μίας κυρτής ακολουθίας αντικειμένων σε γραμμικό χρόνο. Στη συνέχεια υλοποιήστε σε C τον αλγόριθμό σας στην περίπτωση που τα αντικείμενά σας είναι αριθμοί τύπου `double`. Η συνάρτησή σας πρέπει να έχει υπογραφή:

```
void dcsort(double a[], int n);
```

όπου `a` είναι ο προς ταξινόμηση πίνακας, και `n` το πλήθος των στοιχείων του `a`. Κατά την επιστροφή ο πίνακας `a` θα πρέπει να περιέχει την ταξινομημένη ακολουθία.

Τέλος, γράψτε γενικό κώδικα για τον αλγόριθμό σας (σε αναλογία με το γενικό κώδικα της `qsort` που παρέχεται από την `stdlib` της C). Η συνάρτησή σας πρέπει, στην περίπτωση αυτή, να έχει υπογραφή:

```
void csort(void *a, size_t n, size_t size, int (*cmp)(const void*, const void*));
```

όπου `a` είναι ο πίνακας που περιέχει (στην είσοδο) την κυρτή ακολουθία αντικειμένων, `n` το πλήθος των αντικειμένων, μεγέθους `size` το καθένα, και `cmp` δείκτης σε συνάρτηση σύγκρισης στοιχείων του πίνακα `a`. Θεωρήστε ότι η συνάρτηση `*cmp` επιστρέφει `-1` αν το πρώτο στοιχείο είναι μικρότερο από το δεύτερο, `0` αν τα δύο στοιχεία είναι ίσα, και `1` αν το πρώτο στοιχείο είναι μεγαλύτερο από το δεύτερο.

Σύνολο μονάδων: 100