Η/Υ 4-32: Δομές Δεδομένων
Φθινόπωρο 2001 – Παναγιώτα Φατούρου

1ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης
Θεωρητικό Μέρος: Δευτέρα 29/10, ώρα 13:00-14:00
Προγραμματιστικό Μέρος: Κυριακή 28/10, ώρα 23:59.
Τρόπος Παράδοσης
Θεωρητικές Ασκήσεις: Τις λύσεις για το θεωρητικό μέρος θα πρέπει να τις φέρετε στο γραφείο μου έως τη Δευτέρα 29/10. Θα είμαι στο γραφείο μου 13:00-14:00. Δεν θα δεχθώ ασκήσεις μετά τις 14:00. Μπορείτε φυσικά να φέρετε τις ασκήσεις σας και νωρίτερα από αυτήν την ημερομηνία αλλά δικαιολογίες όπως «ήρθα την τάδε μέρα και δεν ήσασταν στο γραφείο σας» δεν θα γίνουν δεκτές. Όσοι δεν έχουν καταφέρει να με βρουν πριν τις 29/10 θα πρέπει να φέρουν τις λύσεις τους στο γραφείο μου την προκαθορισμένη μέρα και ώρα.
Προγραμματιστικές Ασκήσεις: Θα πρέπει να στείλετε με e-mail τον κώδικά σας στην διεύθυνση cs432@cs.uoi.gr μέχρι τις 23:59 της Κυριακής 28/10. Επίσης, θα πρέπει να κρατήσετε την άσκησή σας σε κάποιο directory στην περιοχή σας χωρίς να αλλάξετε την ημερομηνία της μέχρι να γίνει η εξέτασή της. Η ημερομηνία τελευταίας εγγραφής της άσκησής σας θα πρέπει να είναι είτε κάποια προηγουμένη ή η αναγραφόμενη ημερομηνία και ώρα παράδοσης. Μην ετοιμάζετε την άσκησή σας την τελευταία στιγμή. Η ημερομηνία και ώρα παράδοσης προβλέπει πιθανά προβλήματα στο σύστημα και η άσκησή σας θα πρέπει να παραδοθεί μέσα σε αυτή την ημερομηνία ανεξάτητα από τέτοιου είδους προβλήματα. Σας υπενθυμίζω πως ο κώδικάς σας θα πρέπει να είναι γραμμένος στη C, να είναι ευανάγνωστος και να έχει σχολιαστεί κατάλληλα.

Θεωρητικό Μέρος

1. Θεωρήστε ένα πίνακα T που περιέχει δέκα αριθμούς τ.ω. T[j] = j, $0 \le j \le 9$. Ποιές είναι οι τιμές των μεταβλητών a,b, και middle στις διαδοχικές κλήσεις της αναδρομικής συνάρτησης BinarySearch (που δίνεται στη συνέχεια), ξεκινώντας με BinarySearch(T[0..9], 3); [5%]

```
function BinarySearch(table T[a..b], key K): integer
/* επιστρέφει την θέση του στοιχείου K στον ταξινομημένο πίνακα T, αν το K είναι στοιχείο
του πίνακα, διαφορετικά –1 */

    if (a > b) return –1
    middle = ⌊(a+b)/2⌋
    if (K = T[middle]) then return middle
    else  if (K < T[middle]) then
              return BinarySearch(T[a...middle-1], K);
    else                                  /* K > T[middle] */
              return BinarySearch(T[middle+1 .. b], K);
```

2. α. Αποδείξτε τις μεταβατικές ιδιότητες και την συμμετρική ιδιότητα των Ο, Ω, Θ.
   β. Θεωρήστε τον ακόλουθο ορισμό για το Θ:
   Έστω f(n) και g(n) δύο συναρτήσεις. Η f(n) είναι της τάξεως Θ(g(n)), αν υπάρχουν θετικοί ακέραιοι c_1, c_2 και n_0 τ.ω. $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$, για όλα τα $n \ge n_0$.
   Αποδείξτε ότι f(n) = Θ(g(n)) αν και μόνο αν f(n) = O(g(n)) και f(n) = Ω(g(n)).
   [5%]

3. Υποθέστε ότι T[0 .. n-1] είναι ένας πίνακας από εγγραφές με τέσσερα πεδία:

```
struct man {
        char first-name[16];
        char last-name[16];
        int height;
        int weight;
};
```

   Έστω ότι κάθε χαρακτήρας χρειάζεται 1 byte και κάθε ακέραιος χρειάζεται 4 bytes για να αποθηκευθεί. Αν ο πίνακας T ξεκινάει από τη διεύθυνση X, ποια είναι η διεύθυνση του (T[i]).weight, όπου $0 \le i \le n-1$; [5%]

4. Αποδείξτε επαγωγικά ότι: [5%]
   a. αν $T_0 = 0$ και $T_n = 2T_{n-1} + 1$, $n > 0$, τότε $T_n = 2^n - 1$,
   b. για κάθε θετικό ακέραιο n, $11^n - 6$ είναι πολλαπλάσιο του 5.

5. Θεωρήστε τη συνάρτηση $f: \mathbb{N} \to \mathbb{N}$ που ορίζεται ως εξής:

   $f(n) = 1$, αν $n=0$

   $f(n) = 2$, αν $n=1$,

   $f(n) = 4f(n-2) + 2^n$, αν $n > 1$.

   Αποδείξτε επαγωγικά ότι για κάθε ακέραιο $n \ge 3$, $f(n) \le 3n*2^{n-2}$. [10%]

6. Βρείτε την τάξη της χρονικής πολυπλοκότητας T(n) των αλγορίθμων: [20%]
   a. Procedure Foo(integer n):

```
for i from 1 to n do
        x = n;
        while (x > 0) do x = x – i;
```

   Σημείωση: Η ποσότητα $H_n = 1 + 1/2 + 1/3 + \ldots 1/n$ ονομάζεται αρμονικός αριθμός και είναι γνωστό ότι $H_n = O(\ln n)$.

   b. Procedure Mystery(integer n):

```
for i from 1 to n-1 do
        for j = i+1 to n do x = x+1;
```

   c. procedure Puzzle(integer n):

```
for i from 1 to n do
        for j from 1 to 10 do
                for k from n to n+5 do x = x+1;
```

7. Χρησιμοποιώντας ψευδοκώδικα, γράψτε μια διαδικασία η οποία να παίρνει ως παράμετρο μια λίστα από λίστες και να επιστρέφει μια λίστα στην οποία έχουν συγχωνευτεί όλες οι λίστες στοιχεία της λίστας παραμέτρου. Εξηγήστε τον κώδικά σας. [20%]

8. Μια ουρά με δύο άκρα ή διπλή ουρά είναι μια ουρά αναμονής, όπου οι εισαγωγές και εξαγωγές επιτρέπονται και από τα δύο άκρα της ουράς. Οι λειτουργίες που υποστηρίζονται από μια διπλή ουρά είναι:
   - η εισαγωγή στοιχείου στο τέλος της ουράς,
   - η εξαγωγή στοιχείου στην αρχή της ουράς,
   - η εισαγωγή στοιχείου στην αρχή της ουράς, και
   - η εξαγωγή στοιχείου στο τέλος της ουράς.

   Χρησιμοποιώντας ψευδοκώδικα, γράψτε διαδικασίες που να υλοποιούν τις παραπάνω λειτουργίες για τις εξής δύο περιπτώσεις:

   α. σειριακή ουρά, και
   β. συνδεδεμένη ουρά.

   [30%]

Προγραμματιστικό Μέρος

1. Υλοποιήστε στη C την διπλή ουρά της θεωρητικής άσκησης 8. [35%]

2. Θεωρήστε λίστα της οποίας το κάθε στοιχείο είναι μια εγγραφή με 3 στοιχεία: (1) έναν ακέραιο id, (2) ένα θετικό ακέραιο distance, και (3) ένα δείκτη στο επόμενο στοιχείο της λίστας. Ένα στοιχείο της λίστας καθορίζεται απλά από το id του.
   α. Υλοποιήστε στη C τις ακόλουθες διαδικασίες μιας συνδεδεμένης λίστας:
      - Εισαγωγή ενός στοιχείου στη λίστα (μετά την εισαγωγή η κορυφή της λίστας δείχνει στο νέο αυτό στοιχείο).
      - Διαγραφή ενός στοιχείου από τη λίστα (το στοιχείο αναζητείται στη λίστα και αν βρεθεί διαγράφεται από τη λίστα).

   β. Υλοποιήστε στη C τη διαδικασία της θεωρητικής άσκησης 7.

   γ. Επιλύστε προγραμματιστικά το ακόλουθο πρόβλημα στη λίστα. Δεδομένου ενός id εξετάστε αν υπάρχει στοιχείο με τέτοιο id στη λίστα. Αν όχι, ο κώδικάς σας πρέπει να τυπώνει απλά ένα μήνυμα. Αν το στοιχείο υπάρχει στη λίστα (έστω ότι είναι το e), το πρόγραμμά σας θα πρέπει να βρίσκει και να επιστρέφει το id και το distance του στοιχείου που προηγείται του e κατά (e.distance) θέσεις στη λίστα. Γράψτε C κώδικα που να επιλύει το παραπάνω πρόβλημα χρησιμοποιώντας:
   i) δύο δείκτες που μπορούν να κινηθούν μόνο προς τα εμπρός στη λίστα,
   ii) μια ``zig-zag'' διάσχιση χρησιμοποιώντας μια στοίβα από δείκτες, και
   iii) τη μέθοδο αναστροφής δεικτών λίστας. [65%]

ERROR: syntaxerror
OFFENDING COMMAND: --%ztokenexec_continue--

STACK:

false