Κατ’οίκον Εργασία 4

Ημερομηνία Παράδοσης: 2/12/2010

<u>Άσκηση 1</u> [25 μονάδες]

Θεωρήστε τον πιο κάτω πίνακα.

	21	3	5	10	17	8	1	11	2	15

Να εφαρμόσετε διαδοχικά τις πιο κάτω διαδικασίες στον πίνακα.

(α) BuildHeap( ) *(δείξτε όλα τα ενδιάμεσα στάδια)*

(β) DeleteMin( ) *(δείξτε το τελικό αποτέλεσμα)*

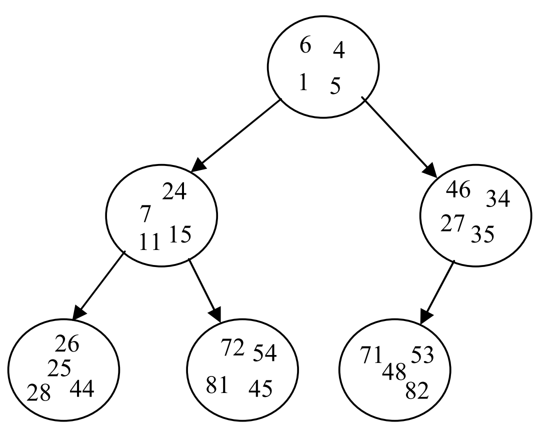
(γ) IncreaseKey (4, 8) [Αύξησε το κλειδί που βρίσκεται στη θέση 4 κατά 8 και αναδιοργάνωσε τη σωρό κατάλληλα.] *(δείξτε το τελικό αποτέλεσμα)*

(δ) Insert (4) *(δείξτε το τελικό αποτέλεσμα)*

(ε) Insert (7) *(δείξτε το τελικό αποτέλεσμα)*

<u>Άσκηση 2</u> [40 μονάδες]

Μία *πολλαπλή ουρά προτεραιότητας βαθμού p* ορίζεται ως μια ακολουθία στοιχείων όπου, κατά την πράξη εισαγωγής (insert), εισάγονται *p* στοιχεία και, κατά την πράξη εξαγωγής ελαχίστου (deletemin), εξάγονται τα *p* μικρότερα στοιχεία. Ένας *πολλαπλός σωρός βαθμού p* είναι μια δομή δεδομένων για υλοποίηση του ΑΤΔ *πολλαπλή ουρά προτεραιότητας βαθμού p* που επεκτείνει τη δομή δεδομένων *σωρός* ως εξής: Ένας πολλαπλός σωρός βαθμού *p* είναι ένα πλήρες δένδρο, όπου, κάθε κόμβος του δένδρου περιέχει *p* στοιχεία, και σε κάθε κόμβο τα *p* στοιχεία του κόμβου είναι μικρότερα από τα στοιχεία των παιδιών του. Επίσης, σε κάθε κόμβο, αν ο κόμβος έχει δύο παιδιά, τότε το ένα από τα παιδιά (είτε το αριστερό είτε το δεξιό) έχει όλα στοιχεία μικρότερα από όλα τα στοιχεία στο άλλο παιδί. Ένα παράδειγμα πολλαπλού σωρού βαθμού *p* παρουσιάζεται στο πιο κάτω παράδειγμα.



Να παρουσιάσετε υλοποίηση ενός πολλαπλού σωρού βαθμού 4 χρησιμοποιώντας πίνακες. Συγκεκριμένα να γράψετε μια καθαρή περιγραφή των δομών που χρειάζονται και την υλοποίηση των πράξεων `Insert4(Q,`x_1, x_2 x_3, x_4`)` (εισήγαγε τα στοιχεία x_1, x_2 x_3, x_4 στον σωρό Q) και `DeleteMin4(Q)` (διέγραψε τα 4 μικρότερα στοιχεία του σωρού Q. Ποιος ο χρόνος εκτέλεσης των διαδικασιών σας;

<u>Άσκηση 3</u> [35 μονάδες]

Σας δίνεται ένα σύνολο $\boldsymbol{C} = \{P_1, P_2, \ldots, P_n\}$ από ιστοσελίδες. Οι ιστοσελίδες αυτές περιέχουν συνδέσμους (links) μεταξύ τους. Μια ιστοσελίδα *A* είναι *προσβάσιμη* από κάποια άλλη ιστοσελίδα *B* αν η μετακίνηση από την ιστοσελίδα *B* προς την ιστοσελίδα *A* είναι δυνατή μέσω μίας ακολουθίας από συνδέσμους. Τέλος, μια ιστοσελίδα ονομάζεται *κύρια* ιστοσελίδα εάν όλες οι υπόλοιπες ιστοσελίδες είναι προσβάσιμες από αυτήν. Θέλουμε να εντοπίσουμε τις κύριες ιστοσελίδες του συνόλου ***C***.

(α) Να προτείνετε τρόπο μοντελοποίησης του προβλήματος χρησιμοποιώντας κατευθυνόμενους γράφους υλοποιημένους με πίνακες γειτνίασης και να παραθέσετε τις σχετικές εγγραφές

(β) Να προτείνετε αποδοτικές υλοποιήσεις για τις πιο κάτω συναρτήσεις χρησιμοποιώντας υλοποίηση γράφων με ***πίνακες γειτνίασης***.

(i) Reachable(A, B, graph G): Η συνάρτηση αυτή θα πρέπει να αποφασίζει κατά πόσο η ιστοσελίδα B είναι προσβάσιμη από την ιστοσελίδα A και, αν ναι, να επιστρέφει τη συντομότερη ακολουθία των ιστοσελίδων από τις οποίες θα πρέπει κάποιος να περάσει για να φθάσει από την A στη B.

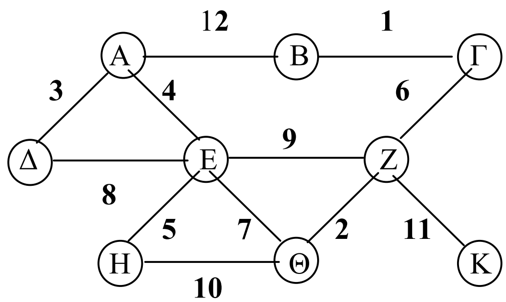
(ii) MasterPage(graph G): Η συνάρτηση αυτή θα πρέπει να βρίσκει και να τυπώνει όλες τις κύριες ιστοσελίδες του γράφου.

Κατ’οίκον Εργασία 5

Ημερομηνία Παράδοσης: 2/12/2010

<u>Άσκηση 1</u> [50 μονάδες]

Έστω ένας μη κατευθυνόμενος γράφος με βάρη G και δύο κορυφές του A και B. Για οποιοδήποτε μονοπάτι ανάμεσα στις A και B ονομάζουμε ως *μήκος μποτιλιαρίσματος* του μονοπατιού το μέγιστο βάρος ανάμεσα σε όλες τις ακμές του. Για παράδειγμα το μήκος μποτιλιαρίσματος του μονοπατιού ΑΔΕΗΘΖΓΒ είναι ίσο με 10. *Απόσταση μποτιλιαρίσματος* ανάμεσα στις A και B ονομάζουμε το ελάχιστο μήκος μποτιλιαρίσματος ανάμεσα σε όλα τα μονοπάτια που συνδέουν τις A και B. Στο παράδειγμα, η απόσταση μποτιλιαρίσματος ανάμεσα στις A και B είναι ίσο με 7 και προέρχεται από το μονοπάτι ΑΕΘΖΓΒ.



Να δώσετε αλγόριθμο ο οποίος, με δεδομένο εισόδου γράφο G υλοποιημένο με λίστες γειτνίασης, να υπολογίζει την απόσταση μποτιλιαρίσματος ανάμεσα σε κάθε ζεύγος κορυφών του. Μπορείτε να υποθέσετε ότι οποιεσδήποτε δύο ακμές του γράφου έχουν διαφορετικό βάρος.

<u>Άσκηση 2</u> [50 μονάδες]

Έστω η συνάρτηση κατακερματισμού $h(x) = x \bmod 10$. Να δείξετε το αποτέλεσμα της διαδοχικής εισαγωγής των κλειδιών 12, 64, 14, 50, 85, 20, 30 για τις ακόλουθες περιπτώσεις δείχνοντας τον πίνακα μετά από κάθε μια από τις εισαγωγές.

(α) Πίνακας κατακερματισμού με αλυσίδωση.

(β) Πίνακας κατακερματισμού με γραμμική αναζήτηση ανοικτής διεύθυνσης και συνάρτηση f όπου $f(i) = (2i+5) \bmod 10$.

(γ) Πίνακας κατακερματισμού με δευτεροβάθμια αναζήτηση ανοικτής διεύθυνσης και συνάρτηση f όπου $f(x,i) = (h(x) + 2i^2 + 1) \bmod 10$.

(δ) Πίνακας διπλού κατακερματισμού ανοικτής διεύθυνσης με την ακόλουθη δεύτερη συνάρτηση κατακερματισμού

$$h_2(x) = 7 - x \bmod 7$$

Να παρουσιάσετε τα αποτελέσματα του επανακατακερματισμού (rehashing) για κάθε μια από τις πιο πάνω περιπτώσεις σε πίνακα μεγέθους 19.

<u>Άσκηση 3</u> [Bonus: 40 μονάδες]

Βρίσκεστε μπροστά από n κλειδωμένες πόρτες P_1, …, P_n, τις οποίες θέλετε να ξεκλειδώσετε. Ο θυρωρός έχει στην διάθεσή του τα n κλειδιά που αντιστοιχούν στις n πόρτες, τα οποία είναι αριθμημένα ως K_1, …, K_n. Τα κλειδιά αυτά είναι όμοια μεταξύ τους σε σχήμα και διαφέρουν μόνο στο μέγεθος. Ανά πάσα στιγμή ο θυρωρός είναι διατεθειμένος να σας δίνει ένα κλειδί το οποίο μπορείτε να χρησιμοποιήσετε για να ελέγξετε κατά πόσο ταιριάζει σε μια από τις κλειδαριές. Κατά τον έλεγχο αυτό μπορείτε να συμπεράνετε ότι το κλειδί ταιριάζει στην κλειδαριά, και να ξεκλειδώσετε την πόρτα, ή ότι είναι μικρότερο ή μεγαλύτερο από ότι ταιριάζει σε αυτήν. Εφόσον επιστρέψετε το κλειδί μπορείτε να του ζητήσετε να σας δώσει κάποιο άλλο από τα κλειδιά. Επομένως, δεν μπορείτε να συγκρίνετε (ή να ταξινομήσετε) μεταξύ τους τα κλειδιά ούτε και τις κλειδαριές.

(α) Να προτείνετε αλγόριθμο με χρόνο εκτέλεσης μέσης περίπτωσης της τάξης $O(n \lg n)$ ο οποίος να λύνει το πρόβλημα.

(β) Είναι δυνατό να λυθεί το πρόβλημα σε χρόνο μικρότερης τάξης από την $\Theta(n \lg n)$;