

Σχεδιασμός Βάσεων Δεδομένων

Διδάσκων: Ιωάννης Κωτίδης

Εαρινό εξάμηνο 2019-2020

Πρώτη Σειρά Ασκήσεων

Ανάθεση: 22-03-2019

Παράδοση: 31-03-2019 Ώρα (23:55)

Οδηγίες

- Η πρώτη σειρά ασκήσεων είναι **ατομική** και **υποχρεωτική**.
- Η υποβολή της εργασίας πρέπει να γίνει στο *eclass*.
- Το παραδοτέο σας θα πρέπει να είναι ένα αρχείο PDF με όνομα *AM.pdf* (όπου AM είναι ο αριθμός μητρώου σας. π.χ. "3170001.pdf").
- Τα διαγράμματα πρέπει να είναι κατασκευασμένα σε κάποιο πρόγραμμα (της επιλογής σας) και όχι σκαναρισμένα χειρόγραφα.
- Πιθανή αντιγραφή θα τιμωρείται με μηδενισμό όλων των εμπλεκομένων.
- Για την επίλυση των ασκήσεων να μελετήσετε τις διαφάνειες των διαλέξεων του μαθήματος.

Άσκηση 1 [μονάδες 3]

Ένα αρχείο περιέχει 20000 εγγραφές φοιτητών σταθερού μήκους. Κάθε εγγραφή αποτελείται από τα εξής πεδία: ΑριθμόςΜητρώου (8 bytes), Επώνυμο (20 bytes), Όνομα (20 bytes), Διεύθυνση (40 bytes), Τηλέφωνο (9 bytes), ΗμερομηνίαΓέννησης (8 bytes), Κωδικό Τμήματος (4 bytes), Κατεύθυνση (4 bytes). Το αρχείο είναι αποθηκευμένο σε ένα δίσκο με τα εξής χαρακτηριστικά (δεν πρόκειται για πραγματικό δίσκο):

- Μέγεθος block 512 bytes
- Πλήθος block ανά ίχνος = 20
- 400 ίχνη ανά επιφάνεια
- 15 πλακέτες διπλής όψης
- Ταχύτητα περιστροφής 2400 rpm
- Μέσος Χρόνος Μετακίνησης Κεφαλής = 30 ms
- Υποθέστε ότι ο χρόνος μετακίνησης στο επόμενο ίχνος είναι μηδενικός.

Δεδομένου ότι κάθε εγγραφή αποθηκεύεται ολόκληρη σε ένα block ζητείται να υπολογίσετε:

α) Τον αριθμό των εγγραφών που χωράνε σε ένα block.

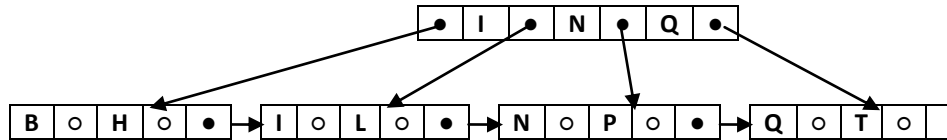
β) Τον συνολικό αριθμό των απαιτούμενων block για την αποθήκευση του αρχείου.

γ) Τον μέσο χρόνο που απαιτείται για την εύρεση μιας εγγραφής αν i) τα μπλοκ του αρχείου αποθηκεύονται συνεχόμενα ii) τα μπλοκ δεν αποθηκεύονται συνεχόμενα.

δ) Υποθέτοντας ότι το αρχείο είναι ταξινομημένο ως προς το πεδίο ΑριθμόςΜητρώου, υπολογίστε τον χρόνο που απαιτείται για την αναζήτηση μιας εγγραφής όταν δίνεται ο ΑριθμόςΜητρώου και χρησιμοποιείται δυαδική αναζήτηση.

Άσκηση 2 [μονάδες 3]

Δίνεται το παρακάτω B+ δένδρο με μέγιστο αριθμό **τριών** κλειδιών (**n=3**) ανά κόμβο/φύλλο το οποίο έχει προκύψει κατόπιν εισαγωγής των τιμών: B,H,I,L,N,P,Q,T. Η εισαγωγή των τιμών έχει γίνει με την σειρά που εμφανίζονται.



☒ Δείκτης προς κόμβο του δένδρου

☐ Δείκτης δεδομένων

☐ Δείκτης δένδρου με τιμή NULL

Ζητείται:

α) Να εισαγάγετε με την σειρά τις τιμές G,A,D,C. Σε κάθε εισαγωγή να παρουσιάσετε τη νέα μορφή του δένδρου και να εξηγήσετε πως ακριβώς προέκυψε.

β) Στο δένδρο που θα προκύψει μετά την εισαγωγή των τιμών του παραπάνω ερωτήματος πόσοι και ποιοι κόμβοι πρέπει να προσπελαστούν για να ανακτηθούν όλες οι εγγραφές με κλειδί αναζήτησης (key) μεγαλύτερο ή ίσο του "C" και μικρότερο ή ίσο του "P" ($key \geq "C"$ AND $key \leq "P"$).

Άσκηση 3 [μονάδες 4]

Ένας πίνακας περιέχει εγγραφές προϊόντων. Το πρωτεύον κλειδί του πίνακα είναι ο αριθμός Barcode του προϊόντος (B#). Ο πίνακας περιέχει 10 εγγραφές με τις παρακάτω τιμές για το B#.

Record	B#
R1	48
R2	32
R3	64
R4	53
R5	87
R6	29
R7	16
R8	65
R9	70
R10	49

Έστω ένα αρχείο ευρετηρίου που χρησιμοποιεί την μέθοδο του γραμμικού κατακερματισμού με αρχικό μέγεθος **2 buckets (m=1)** χωρητικότητας δύο εγγραφών έκαστο. Για την κατανομή των τιμών χρησιμοποιούνται τα $i=1$ λιγότερο σημαντικά bits, ενώ η συνάρτηση κατακερματισμού είναι η **$h(B) = B \bmod 15$** . Το m πρέπει να αυξάνεται όταν το utilization του ευρετηρίου γίνει μεγαλύτερο ή ίσο του **80%**. Το i αυξάνεται μόνο όταν κρίνεται απαραίτητο. Επίσης, δεν υπάρχει όριο στον αριθμό σελίδων υπερχειλίσης. Κάθε σελίδα υπερχειλίσης χωράει και αυτή από δύο εγγραφές.

Ζητείται:

α) Να εισαγάγετε τα κλειδιά των παραπάνω εγγραφών με την σειρά που σας δίνονται στον παραπάνω πίνακα. Να εμφανίστε την μορφή του ευρετηρίου μετά από κάθε εισαγωγή κλειδιού δείχνοντας και όσα ενδιάμεσα βήματα απαιτούνται. Κάθε πράξη εισαγωγής πρέπει να εκτελείται στο αποτέλεσμα της προηγούμενης και όχι στο αρχικό ευρετήριο.

β) Να υπολογίσετε τον μέσο αριθμό προσπελάσεων που απαιτούνται για την ανάκτηση μιας εγγραφής με βάση την τιμή του κλειδιού B# για κάθε μία από τις παρακάτω δύο περιπτώσεις:

- I. Η τιμή του κλειδιού B# υπάρχει στο ευρετήριο.
- II. Η τιμή του κλειδιού B# δεν υπάρχει στο ευρετήριο.

Προς διευκόλυνσή σας ακολουθεί η εισαγωγή του πρώτου κλειδιού:

Εισαγωγή Κλειδιού 48: $h(48) = 48 \bmod 15 = 3$ (0011)

	48 (0011)

0

1

Utilization=(1/4)=25%

Άσκηση 4 [bonus μονάδες 2]

Ένα στατικό αρχείο κατακερματισμού έχει αρχικά 600 κάδους στην κύρια περιοχή και κάθε κάδος χωράει σε μία σελίδα. Μετά την εισαγωγή ενός αριθμού εγγραφών δημιουργείται μια περιοχή υπερχειλίσας 600 σελίδων. Υπάρχει δυνατότητα αναδιοργάνωσης του αρχείου έτσι ώστε να εξαλειφτούν οι σελίδες υπερχειλίσας. Το κόστος για την αναδιοργάνωση του αρχείου είναι **μόνο** το κόστος για την μεταφορά (διάβασμα και γράψιμο) όλων των σελίδων του αρχείου. Η μόνη λειτουργία που εκτελείται περιοδικά στο αρχείο είναι η λειτουργία τυχαίας αναζήτησης **fetch “βρες και φέρε”**.

Υποθέστε ότι ο δίσκος στον οποίο βρίσκεται το αρχείο έχει τα παρακάτω χαρακτηριστικά:

- Μέσος χρόνος μετακίνησης κεφαλής $s=16\text{ms}$
- Μέση καθυστέρηση περιστροφής $rd=8,3\text{ms}$
- Χρόνος μεταφοράς του block $btt=1\text{ms}$

Έστω ότι θέλουμε να εκτελέσουμε με επιτυχία (η τιμή του κλειδιού υπάρχει στο αρχείο) N τυχαίες αναζητήσεις. Έχουμε τις εξής δυνατότητες:

α) Να εκτελέσουμε τις N αναζητήσεις στο αρχείο ως έχει.

β) Να αναδιοργανώσουμε το αρχείο και στην συνέχεια να εκτελέσουμε τις N αναζητήσεις.

Ζητείται να προσδιορίσετε την ελάχιστη τιμή του N ώστε να συμφέρει η αναδιοργάνωση του αρχείου (περίπτωση β). Δηλαδή το κόστος για την αναδιοργάνωση του αρχείου προστιθέμενο με το κόστος της αναζήτησης μετά την αναδιοργάνωση, θα πρέπει να είναι μικρότερο από το κόστος της αναζήτησης πριν την αναδιοργάνωση του αρχείου.