

# Βάσεις Δεδομένων II

## Εργασία 01

**[A]. 1.** Το μεγαλύτερο πλήθος διαγραφών που μπορούν να γίνουν είναι 7.

Διαγράφουμε το 1\*. Δεν συντελείται κάποια αλλαγή στο δέντρο.

Διαγράφουμε το 2\*. Πάλι το ίδιο, το δέντρο παραμένει ως έχει.

Διαγράφουμε το 5\*. Μένει το 6\* μόνο του και επειδή έχει πέσει κάτω από το 50% γίνεται συγχώνευση με το δεξί του αδέρφι αφού ανακατανομή δεν μπορεί να γίνει και έτσι ο νέος κόμβος είναι (6\*,8\*,10\*). Διαγράφεται το 8 από τον γονιό.

Διαγράφουμε το 6\*. Δεν γίνεται κάποια αλλαγή στο δέντρο.

Διαγράφουμε το 8\*. Όπως και προηγουμένως το 10\* θα συγχωνευθεί με το δεξί του αδέρφι και ο νέος κόμβος θα είναι ο (10\*,18\*,27\*). Θα διαγραφεί και το 18 από τον γονιό. Ο γονιός τώρα είναι ο (32, 40).

Διαγράφουμε το 10\*. Δεν γίνεται κάποια αλλαγή.

Διαγράφουμε το 52\*. Ο κόμβος πέφτει κάτω από το 50% και έτσι το 58 θα συγχωνευθεί με το δεξί του αδέρφι. Ο νέος κόμβος θα είναι ο (58\*,73\*,80\*). Θα διαγραφεί το 73 από τον γονιό όμως αυτό θα προκαλέσει συγχώνευση με το αριστερό αδέρφι του γονιού. Η ρίζα επειδή δεν υπάρχει λόγος να υπάρχει πλέον θα κατέβει και αυτή κάτω στον κόμβο και έτσι ο νέος κόμβος-ρίζα με 5 παιδιά, θα είναι ο (32, 40, 50, 85).

**[A]. 2.** Το μεγαλύτερο πλήθος εισαγωγών που πρέπει να γίνουν για να γεμίσει ο κόμβος είναι 11.

Εισάγουμε το 53\*. Αυτό θα μπει στον αριστερά κόμβο (έκτος κατά σειρά από αριστερά προς τα δεξιά) και δεν θα συμβεί τίποτε άλλο.

Εισάγουμε το 60\*. Και αυτό θα μπει στον αριστερά κόμβο και θα γεμίσει ο κόμβος.

Εισάγουμε το 75\*. Αυτό θα μπει στον μεσαίο κόμβο και δεν θα γίνει κάτι άλλο.

Εισάγουμε το 79\*. Θα μπει στον μεσαίο κόμβο και θα γεμίσει.

Εισάγουμε το 85\* και μπαίνει στον δεξιά κόμβο χωρίς να γίνει κάτι άλλο.

Εισάγουμε το 86\* και μπαίνει στον δεξιά κόμβο με αποτέλεσμα και αυτός ο κόμβος να γεμίσει.

Εισάγουμε το 87\*. Θα υπερχειλίσει ο δεξιά κόμβος και θα διασπαστεί στους κόμβους (85\*,86\*,87\*) και (91\*,99\*). Επίσης θα ανέβει στον γονιό η τιμή 91.

Εισάγουμε το 88\*. Θα μπει στον προτελευταίο κόμβο και θα γίνει (85\*,86\*,87\*,88\*).

Εισάγουμε το 92\*. Δεν γίνεται κάτι άλλο.

Εισάγουμε το 93\*. Τώρα ο δεξιά κόμβος γεμίζει.

Εισάγουμε το 94\*. Ο δεξιά κόμβος υπερχειλίζει και διασπάτε στους κόμβους (91\*,92\*,93\*) και (94\*,99\*). Επίσης ανεβαίνει η τιμή 94 στον γονιό.

Σε αυτό το σημείο ο γονιός είναι γεμάτος και οι τιμές ευρετηρίου που έχει είναι (73, 85, 91, 94).

**[B]. 1.** Στο επίπεδο των φύλλων (καταχωρίσεις δεδομένων – επίπεδο 1) επειδή έχουν πληρότητα 100% και επειδή ο συνολικός αριθμός των εγγραφών είναι 100.000, θα έχουμε συνολικά  $100.000/10 = 10.000$  κόμβους-φύλλα.

Στο επίπεδο 2 επειδή η πληρότητα είναι 50% κάθε κόμβος ευρετηρίου θα χωράει 5 τιμές και 6 pointers. Το ίδιο θα συμβαίνει και στα πιο πάνω επίπεδα. Οπότε θα χρειαστούμε  $\text{ceil}(10.000/6) = 1.667$  κόμβους για το επίπεδο 2.

Για το επίπεδο 3 θα έχουμε  $\text{ceil}(1.667/6) = 278$  κόμβους ευρετηρίου.

Για το επίπεδο 4 θα έχουμε  $\text{ceil}(278/6) = 47$  κόμβους.

Για το επίπεδο 5 θα είναι  $\text{ceil}(47/6) = 8$  οι κόμβοι.

Για το επίπεδο 6 θα είναι  $\text{ceil}(8/6) = 2$  κόμβοι.

Τέλος στο επίπεδο 7, θα είναι η ρίζα με τα 2 παιδιά της.

**[Β]. 2.** Επειδή οι κόμβοι θα είναι κατά 70% γεμάτοι σε όλα τα επίπεδα, κάθε κόμβος θα αποθηκεύει  $\text{ceil}(1000/100 \cdot 0,7) = 7$  καταχωρίσεις (ευρετηρίου ή δεδομένων) και 8 pointers για το εσωτερικό δέντρο. Οπότε στο επίπεδο 1 θα έχουμε  $\text{ceil}(100.000/7) = 14.286$  κόμβους-φύλλα.

Στο επίπεδο 2 θα έχουμε  $\text{ceil}(14.286/8) = 1.786$  κόμβους ευρετηρίου.

Στο επίπεδο 3 θα έχουμε  $\text{ceil}(1.786/8) = 224$  κόμβους.

Στο επίπεδο 4 θα έχουμε  $\text{ceil}(224/8) = 28$  κόμβους.

Στο επίπεδο 5 θα έχουμε  $\text{ceil}(28/8) = 4$  κόμβους.

Τέλος στο επίπεδο 6 θα βρίσκεται η ρίζα με τα 4 παιδιά.

**[Γ].** Ο τύπος για να βρούμε τον αριθμό των περασμάτων που απαιτούνται για την ταξινόμηση είναι  $1 + \text{ceil}(\log_{B-1}(N/B))$  όπου  $B$  είναι τα διαθέσιμα πλαίσια και  $N$  είναι ο αριθμός των σελίδων που καταλαμβάνει το αρχείο στον δίσκο. Με αυτά τα δεδομένα θα έχουμε:

**[Γ]. 1.**  $B = 3$ . Θα χρειαστούν  $1 + \text{ceil}(\log_2 \text{ceil}(20.000/3)) = 1 + \text{ceil}(\log_2 6.667) = 1 + 3 = 4$  περάσματα.

**[Γ]. 2.**  $B = 101$ . Θα χρειαστούν  $1 + \text{ceil}(\log_{100} \text{ceil}(20.000/101)) = 1 + \text{ceil}(\log_{100} 199) = 1 + 2 = 3$  περάσματα.

**[Γ]. 3.**  $B = 142$ . Θα χρειαστούν  $1 + \text{ceil}(\log_{141} \text{ceil}(20.000/142)) = 1 + \text{ceil}(\log_{141} 141) = 1 + 1 = 2$  περάσματα.

**[Γ]. 4.**  $B = 19.999$ . Θα χρειαστούν  $1 + \text{ceil}(\log_{19.998} \text{ceil}(20.000/19.999)) = 1 + \text{ceil}(\log_{19.998} 2) = 1 + 1 = 2$  περάσματα.

**[Γ]. 5.**  $B = 20.000$ . Θα χρειαστούν  $1 + \text{ceil}(\log_{19.999} \text{ceil}(20.000/20.000)) = 1 + \text{ceil}(\log_{19.999} 1) = 1 + 0 = 1$  περάσματα. Είναι λογικό το αποτέλεσμα αφού

χωράει όλος ο πίνακας στην ενδιάμεση μνήμη. Εκτελείται ο αλγόριθμος ταξινόμησης και αμέσως αποθηκεύεται ταξινομημένος ο πίνακας στον δίσκο.

---

**Ονοματεπώνυμο: Παναγιώτης Γιαννουτάκης**

**ΑΜ: 38/12**

**e-mail: [it1238@uom.edu.gr](mailto:it1238@uom.edu.gr)**