

## 2η Σειρά Ασκήσεων

Διογένη Τσολάκου

3170164

### Άσκηση 1

Η σχέση R περιέχει 45000 εγγραφές, οι οποίες είναι αποθηκευμένες σε block/σελίδες χωρητικότητας 100 εγγραφών. Άρα συνολικά έχουμε 450 block/σελίδες. Η μνήμη έχει μέγεθος 4 block άρα μπορεί να αποθηκεύσει έως 400 εγγραφές σε κάθε μία στιγμή. Συνολικά θα έχουμε  $450/4 = 112.5$ , άρα 113 ταξινομημένες λίστες.

1.

2. Θα χρειαστούν συνολικά  $B(R) = \lceil ((M-1)^{k-1}) * M \rceil$ , όπου  $B(R)$  είναι οι 450 σελίδες και  $k$  τα περάσματα για την ταξινόμηση της σχέσης R. Δηλαδή  $450 = \lceil ((3)^{k-1}) * 4 \rceil > 112.5 = \lceil 3^{k-1} \rceil$ . Το  $k$  βγαίνει 5.299, άρα χρειαζόμαστε  $\text{ceiling}(5.299) = 6$  περάσματα.

3. Το συνολικό κόστος I/O θα είναι  $2 * (k-1) * B(R) = 2 * 5 * 450 = 4500$ .

4. Για να γίνει η ταξινόμηση της σχέσης σε δύο μόνο περάσματα το μέγεθος της μνήμης πρέπει να είναι  $M \geq \sqrt{B(R)}$ . Επομένως θέλουμε  $M \geq \sqrt{450} \rightarrow M \geq 21.213$ . Άρα χρειαζόμαστε μνήμη μεγέθους 22 block.

### Άσκηση 2

Έχουμε  $B(R1) = 15000/100 = 150$  και  $B(R2) = 9000/100 = 90$ .

1. Με το Sort Merge Join έχουμε κόστος  $5 * (B(R1) + B(R2)) = 5 * (150 + 90) = 5 * (240) = 1200$  I/Os.

Με τον Nested Loop Join έχουμε κόστος  $B(R1) + T(R1) * B(R2)$

$= 150 + 15000 \cdot 90 = 150 + 1350000 = 1350150$  I/Os με την απλοϊκή λύση και  $B(R1)$   
 $+ \text{ceiling}(B(R1)/(M - 1)) \cdot B(R2) = 150 + \text{ceiling}(150/12) \cdot 90$

$= 150 + \text{ceiling}(12.5) \cdot 90 = 150 + 13 \cdot 90 = 150 + 1170 = 1320$  σελίδες με την  
βελτιωμένη λύση επιλέγοντας ως εξωτερική την  $B(R1)$ . Επιλέγοντας ως εξωτερική  
την  $B(R2)$  έχουμε  $B(R2) + \text{ceiling}(B(R2)/(M - 1)) \cdot B(R1)$

$= 90 + \text{ceiling}(90/12) \cdot 150 = 90 + \text{ceiling}(7.5) \cdot 150 = 90 + 8 \cdot 150 = 90 + 1200 = 1290$   
σελίδες.

Με τον Hash Join έχουμε  $3 \cdot (B(R1) + B(R2)) = 720$  I/Os.

Άρα για την εκτέλεση του ερωτήματος θα επιλέξουμε τον αλγόριθμο Hash Join.

**2.** Με το Sort Merge Join έχουμε κόστος  $3 \cdot (B(R1) + B(R2)) = 3 \cdot (150 + 90) =$   
 $3 \cdot (240) = 720$  I/Os. Αυτή τη φορά πολλαπλασιάζουμε επί 3 καθώς η μνήμη είναι  
μεγαλύτερη και εκπληρώνει τις απαιτήσεις μνήμης του βελτιωμένου SMJ.

Ο απλός NLJ δεν αλλάζει. Ο βελτιωμένος NLJ γίνεται

$B(R1) + \text{ceiling}(B(R1)/(M - 1)) \cdot B(R2) = 150 + \text{ceiling}(150/99) \cdot 90$

$= 150 + \text{ceiling}(1.5151) \cdot 90 = 150 + 2 \cdot 90 = 150 + 180 = 330$  I/Os με την  $B(R1)$  ως  
εξωτερική. Με την  $B(R2)$  ως εξωτερική έχουμε

$B(R2) + \text{ceiling}(B(R2)/(M - 1)) \cdot B(R1) = 90 + \text{ceiling}(90/99) \cdot 150$

$= 90 + \text{ceiling}(0.909) \cdot 150 = 90 + 1 \cdot 150 = 240$  I/Os.

Με το Hash Join έχουμε ότι η μία σχέση χωράει στην μνήμη αφού

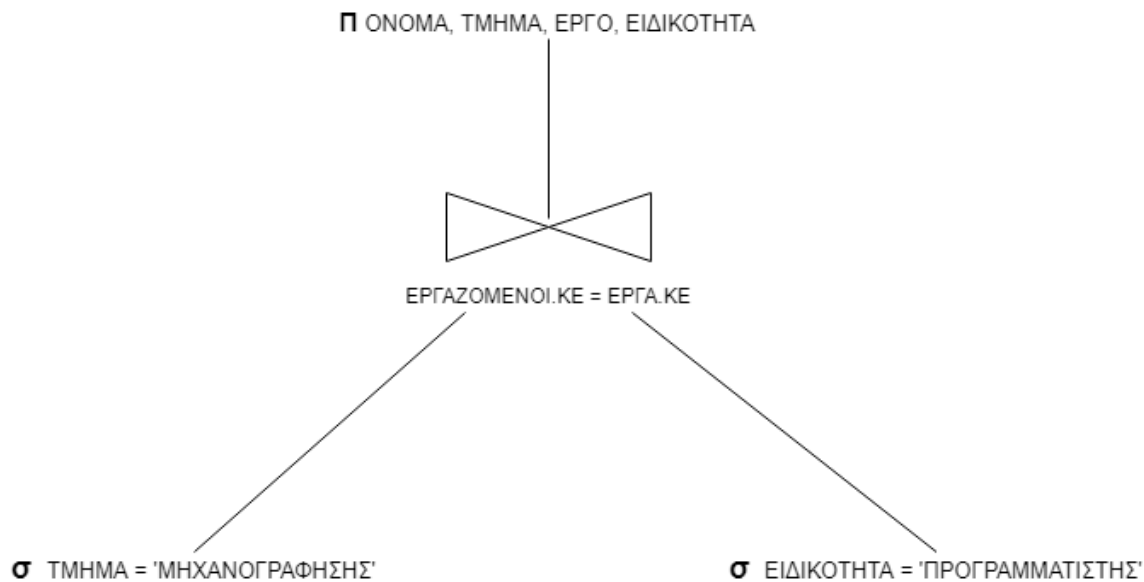
$B(R2) = 90$ . Άρα το κόστος θα είναι  $B(R1) + B(R2) = 150 + 90 = 240$  I/Os.

Άρα μπορούμε να επιλέξουμε πάλι τον αλγόριθμο Hash Join ή τον βελτιωμένο  
Nested Loop Join αφού έχουν ίδιο αριθμό I/Os.

### Άσκηση 3

$B(R1) = 250$  και  $B(R2) = 50$

1.



2. R1 είναι η αριστερή σχέση στο λογικό πλάνο και R2 η δεξιά.

Έχουμε  $T(R1) = 500$  εγγραφές καθώς έχουμε 5000 εγγραφές συνολικά και 10 διαφορετικά τμήματα και με την υπόθεση ότι τα δεδομένα είναι κατανομημένα ομοιόμορφα. Με παρόμοιο τρόπο έχουμε  $T(R2) = 400$  εγγραφές καθώς έχουμε 2000 εγγραφές και 5 ειδικότητες. Για κάθε σελίδα της σχέσης Εργαζόμενοι χωράνε συνολικά

$5000/250 = 20$  εγγραφές, άρα  $B(R1) = 500/20 = 25$ . Ενώ για τη σχέση Έργα έχουμε  $2000/50 = 40$  εγγραφές, άρα  $B(R2) = 400/40 = 10$ .

**α)** Μπορούμε να χρησιμοποιήσουμε τον βελτιωμένο SMJ καθώς πληρούμε τις απαιτήσεις μνήμης  $M \geq \sqrt{B(R1)+B(R2)}$  -->

$M \geq \sqrt{25+10} \rightarrow M \geq \sqrt{35} \rightarrow M \geq 5.91$ .

Άρα έχουμε  $3*(B(R1) + B(R2)) = 3*35 = 105$  I/Os. Επομένως το συνολικό κόστος είναι  $B(R1) + B(R2) + SMJ = 35 + 105 = 140$  I/Os.

**β)** Με τον απλό Nested Loop Join έχουμε  $B(R1) + T(R1)*B(R2)$

$= 25 + 500*10 = 25 + 5000 = 5025$  I/Os.

Με τον βελτιωμένο και την B(R1) ως εξωτερική έχουμε

$$B(R1) + \text{ceiling}(B(R1)/(M - 1)) * B(R2) = 25 + \text{ceiling}(25/7) * 10 \\ = 25 + \text{ceiling}(3.571) * 10 = 25 + 4 * 10 = 25 + 40 = 65 \text{ I/Os.}$$

$$\text{Άρα συνολικά } B(R1) + B(R2) + NLJ = 35 + 65 = 100 \text{ I/Os.}$$

$$\text{Με την B(R2) ως εξωτερική έχουμε } B(R2) + \text{ceiling}(B(R2)/(M - 1)) * B(R1) \\ = 10 + \text{ceiling}(10/7) * 25 = 10 + \text{ceiling}(1.42) * 25 = 10 + 2 * 25 = 10 + 50 \\ = 60 \text{ I/Os. Άρα συνολικά } B(R1) + B(R2) + NLJ = 35 + 60 = 95 \text{ I/Os}$$

Άρα το ελάχιστο κόστος εκτέλεσης σε I/O της επερώτησης είναι 95 με τον βελτιωμένο Nested Loop Join και την B(R2) ως εξωτερική.

#### Άσκηση 4

1. Αφού κάνουμε table scan θα διαβάσουμε όλο τον πίνακα, άρα το I/O είναι 100 σελίδες.

2. Παρόμοια με την (1) διαβάζουμε όλο τον πίνακα άρα το I/O είναι 10 σελίδες.

3. Με την υπόθεση ότι τα δεδομένα είναι ομοιόμορφα κατανεμημένα τότε από το (1) και το (2) έχουμε  $T(1) = 2000 \text{ εγγραφές} / 20 \text{ πόλεις} = 100$  και  $T(2) = 9$  εγγραφές μιας και θέλουμε τα πανεπιστήμια με κατάταξη μικρότερη του 10. Κάθε σελίδα της σχέσης Φοιτητές χωράει  $2000/100 = 20$  εγγραφές και κάθε σελίδα της σχέσης Πανεπιστήμια χωράει  $100/10 = 10$  εγγραφές. Επομένως έχουμε  $B(1) = T(1)/20 = 100/20 = 5$  και  $B(2) = T(2)/10 = 9/10 = 0.9 = 1$ . Μπορούμε να χρησιμοποιήσουμε τον βελτιωμένο SMJ καθώς  $M \geq \text{sqrt}(B(1) + B(2)) \rightarrow M \geq \text{sqrt}(6) \rightarrow M \geq 2.44$ , το οποίο ισχύει στη συγκεκριμένη περίπτωση.  $3 * (B(1) + B(2)) = 3 * 6 = 18 \text{ I/Os.}$

4. Με την υπόθεση ότι τα δεδομένα είναι ομοιόμορφα κατανεμημένα έχουμε  $2000 \text{ φοιτητές} / 100 \text{ πανεπιστήμια} = 20 \text{ φοιτητές σε κάθε πανεπιστήμιο. Σε κάθε πανεπιστήμιο υπάρχει ένας Αθηναίος φοιτητής καθώς έχουμε } 20 \text{ φοιτητές} / 20$

πόλεις = 1 από κάθε πόλη. Άρα έχουμε ότι  $T(3) = 9$  εγγραφές μιας και θέλουμε τους Αθηναίους φοιτητές που φοιτούν στα 9 κορυφαία πανεπιστήμια. Αυτές οι εγγραφές χωράνε σε ένα μόνο block. Επιπλέον, μπορούμε να θεωρήσουμε ότι κάθε φοιτητής θα έχει προεγγραφεί σε 3 ΜΠΣ μιας και έχουμε συνολικά 5 ΜΠΣ, άρα το ευρετήριο θα γυρίζει 3 εγγραφές σε κάθε αναζήτηση.

Άρα έχουμε  $B(3) + T(3)*3 = 1 + 9*3 = 1 + 27 = 28$  I/Os.

5. Από το (4) θεωρήσαμε ότι κάθε φοιτητής έχει προεγγραφεί κατά μέσο όρο σε 3 ΜΠΣ, άρα για τους 9 φοιτητές υπάρχουν 27 εγγραφές. Επομένως θα διαβάζει όλο τον πίνακα αυτών των 27 εγγραφών. Οι 27 εγγραφές χωράνε σε 2 σελίδες καθώς  $2000 \text{ φοιτητές} / 100 \text{ σελίδες} = 20 \text{ εγγραφές}$ . Άρα το I/O ισούται με 2.

6. Η προβολή των συνολικών εγγραφών έχει κόστος 0 I/Os.

Άρα το συνολικό κόστος σε I/O του φυσικού πλάνου θα είναι

$100 + 10 + 18 + 28 + 2 + 0 = 158$  I/Os.