ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Συστήματα Διαχείρισης Βάσεων Δεδομένων

Σειρά Ασκήσεων 2 2023-2024

Ονοματεπώνυμο: Κεχριώτη Ελένη

Άσκηση 1

Δίνονται τα εξής:
R1(a,b,c,d) R2(p,q,r,s,t,u,v)
T(R1)=5000000, B(R1)=50000, V(R1,a)=100, V(R1,b)=50
T(R2)=10000, B(R2)=2000

Q1: SELECT * FROM R1 WHERE a=40 and b=50

Εφόσον υπάρχουν V(R1,a)=100 διακριτές τιμές του γνωρίσματος a και είναι ομοιόμορφα κατανεμημένες η πιθανότητα εμφάνισης εγγραφών με τιμή 40 είναι $\frac{1}{V(R1,a)} = \frac{1}{100}$.

Αντίστοιχα, η πιθανότητα εμφάνισης εγγραφών με τιμή 50 στο γνώρισμα b είναι $\frac{1}{V(R1,b)} = \frac{1}{50}$.

Γνωρίζουμε ότι οι κατανομές των τιμών είναι ανεξάρτητες μεταξύ του, επομένως η πιθανότητα εμφάνισης εγγραφών με τιμή 40 στο γνώρισμα a και ταυτόχρονα τιμή 50 στο γνώρισμα b είναι $\frac{1}{V(R1,a)} \times \frac{1}{V(R1,b)}$.

Συνεπώς, $\frac{1}{100} \times \frac{1}{50} \times T(R1) = 1000$ εγγραφές θα πληρούν την συνθήκη και θα επιστρέφονται ως έξοδο στο επερώτημα.

Το ευρετήριο βρίσκεται στην μνήμη άρα το κόστος σε I/O για τη διάσχιση του ευρετηρίου είναι 0. Επομένως το κόστος θα είναι στην ουσία το κόστος διάσχισης του B+ δέντρου. Το ευρετήριο είναι ένα απλό, non clustered, ευρετήριο άρα

$$\text{cost(Q1)} = \text{T(Q1)} = 1000$$

Q2: SELECT * FROM R2 WHERE s>96 AND s<=220

Λόγω ομοιόμορφης κατανομής, η πιθανότητα εμφάνισης εγγραφών με τιμή >96 και <220 στον πίνακα R2 είναι:

- Από το bucket [1..100] → $4 \times \frac{2500}{100} = 100$.
- Από το bucket [101..200] → 500.
- Από το bucket [201..400] → $20 \times \frac{4000}{200} = 400$.

Άρα οι εγγραφές του πίνακα R2 με τιμές (96,220] στο γνώρισμα s είναι 100+500+400 = 1000. Για την διάσχιση του ευρετηρίου το κόστος είναι 0, όπως προηγουμένως και λόγω του ότι το ευρετήριο είναι απλό το κόστος είναι:

$$\text{cost(Q2)} = \text{T(Q2)} = 1000$$

Β) Από τα δεδομένα της άσκησης έχουμε ότι T(R1)=5000000 και B(R1)=50000, άρα σε μια σελίδα χωράνε 5000000/50000 = 100 εγγραφές της σχέσης R1. Αντίστοιχα, σε μια σελίδα χωράνε 10000/2000 = 5 εγγραφές της σχέσης R2.

Για το Q1, αν το ευρετήριο ήταν ευρετήριο συστάδων τότε το κόστος θα ήταν το B(Q1), όπου

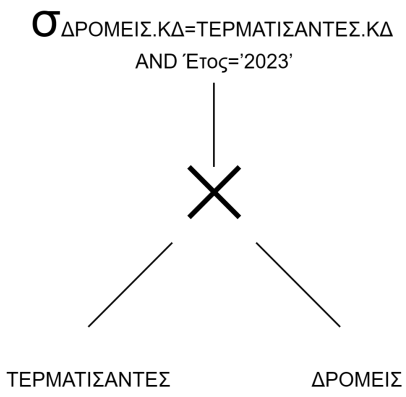
$$B(Q1) = \lceil \frac{T(Q1)}{100} \rceil = \frac{1000}{100} = 10.$$

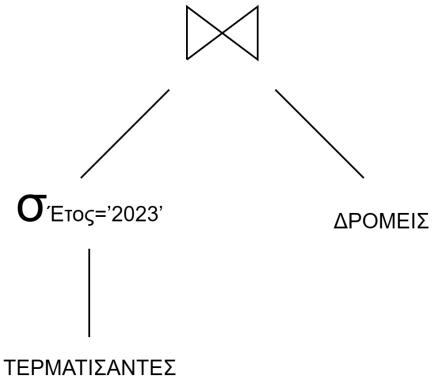
Αντίστοιχα για το Q2, αν το ευρετήριο ήταν ευρετήριο συστάδων τότε το κόστος θα ήταν το B(Q2), όπου

$$B(Q2) = \lceil \frac{T(Q2)}{5} \rceil = \frac{1000}{5} = 200.$$

Άσκηση 2

Α) Αρχικό λογικό πλάνο → Τελικό λογικό πλάνο





Β)
α) Από τα δεδομένα έχουμε:

Τ(ΔΡΟΜΕΙΣ)=40000 και Β(ΔΡΟΜΕΙΣ)=200
Τ(ΤΕΡΜΑΤΙΣΑΝΤΕΣ)=60000 και Β(ΤΕΡΜΑΤΙΣΑΝΤΕΣ)=600

Σε μία σελίδα χωρούν Τ(ΔΡΟΜΕΙΣ)/Β(ΔΡΟΜΕΙΣ) = 40000/200 = 200 εγγραφές της R
Σε μία σελίδα χωρούν Τ(ΤΕΡΜΑΤΙΣΑΝΤΕΣ)/Β(ΤΕΡΜΑΤΙΣΑΝΤΕΣ) = 60000/600 = 100 εγγραφές της ΤΕΡΜΑΤΙΣΑΝΤΕΣ.

Έστω Χ = σΈτος='2023' (ΤΕΡΜΑΤΙΣΑΝΤΕΣ)

Γνωρίζουμε ότι τα έτη είναι 4 και οι τερματίσαντες κατανέμονται ομοιόρφα στα 4 έτη. Συνεπώς,

$T(x) = \frac{T(\text{ΤΕΡΜΑΤΙΣΑΝΤΕΣ})}{V(\text{ΤΕΡΜΑΤΙΣΑΝΤΕΣ, Έτος})} = \frac{60000}{4} = 15000$ και
$B(x) = \frac{15000}{100} = 150$

Άρα το cost(x) = min(600 με table scan, 150 me index seek) = 150 (γιατί είναι clustered το ευρετήριο).

Cost(ΔΡΟΜΕΙΣ) = Β(ΔΡΟΜΕΙΣ) = 200

α) Ο αλγόριθμος SMJ απαιτεί οι πίνακες που συμμετέχουν στη σύζευξη (εδώ οι Χ, ΔΡΟΜΕΙΣ) να είναι ταξινομημένοι. Ο Χ δεν είναι ταξινομημένος, οπότε θα πρέπει να ταξινομηθεί. Για την ακρίβεια είναι ταξινομημένος λόγω του clustered index με βάση το έτος, αλλά όχι με το ΚΑ που χρειαζόμαστε για το join. Οπότε θα χρειαστεί να ταξινομηθεί με βάση το γνώρισμα ΚΑ.

Στην έξοδο της επιλογής του έτους, επιστρέφονται 15000 εγγραφές και κάθε σελίδα χωράει 100 εγγραφές της Χ, άρα θέλουν 150 σελίδες που δεν χωράνε στη μνήμη αφου Μ = 21.

Επομένως δεν μπορεί να γίνει απευθείας ταξινόμηση στην μνήμη του Χ με γνωστούς αλγορίθμους ταξινόμησης (όπως QuickSort, MergeSort κ.λπ).
Ο Χ θα δημιουργήσει ceil(150/21) = 8 υπολίστες

Ο πίνακας ΔΡΟΜΕΙΣ είναι ήδη ταξινομημένος, εφόσον υπάρχει ήδη clustered index στο γνώρισμα ΚΔ, επομένως δεν έχουμε κόστος για την ταξινόμηση του. Για τον πίνακα ΔΡΟΜΕΙΣ θέλουμε 200 σελίδες που δεν χωράνε στην μνήμη αφού Μ = 21
Ο ΔΡΟΜΕΙΣ θα δημιουργήσει ceil(200/21) = 10 υπολίστες
Συνολικά, λοιπόν δημιουργούνται 8 + 10 = 18 < 21 υπολίστες που μπορούν να συγχωνευθούν μαζί στη συνέχεια.

B(X) = 150 > 21
B(ΔΡΟΜΕΙΣ) = 200 > 21
και B(X) + B(ΔΡΟΜΕΙΣ) = 350 < m^2 = 441
Αυτό σημαίνει ότι ο αλγόριθμος θα τρέξει με την αποδοτική έκδοση την δεύτερη φορά.

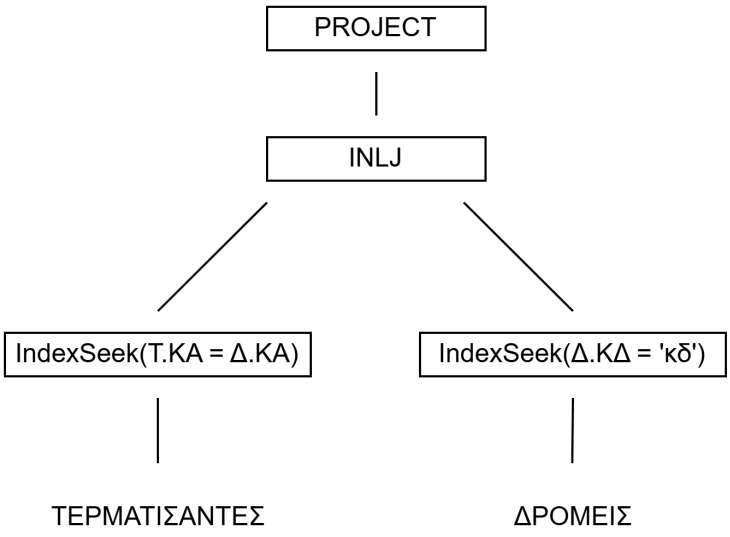
Cost(SMJ) = cost(X) + 2B(X) + cost(ΔΡΟΜΕΙΣ) = 150 + 2* 150 + 200 = 650

β) Με εξωτερική σχέση την Χ:
$NLJ_{X, ΔΡΟΜΕΙΣ}$ = cost(X) + ceil(B(X)/(m-1)) * Cost(ΔΡΟΜΕΙΣ) = 150 + 8*200 = 1750

Με εξωτερική σχέση την ΔΡΟΜΕΙΣ:
$NLJ_{ΔΡΟΜΕΙΣ, X}$ = cost(ΔΡΟΜΕΙΣ) + ceil(B(ΔΡΟΜΕΙΣ)/(m-1)) * Cost(X) = 200 + 10*150 = 1700

Άσκηση 3

```
SELECT *
FROM ΔΡΟΜΕΙΣ, ΤΕΡΜΑΤΙΣΑΝΤΕΣ
WHERE ΔΡΟΜΕΙΣ.ΚΔ=ΤΕΡΜΑΤΙΣΑΝΤΕΣ.ΚΔ AND ΔΡΟΜΕΙΣ.ΚΔ=κδ
```



Θα χρειαστεί να δημιουργήσουμε ένα ευρετήριο πάνω στο γνώρισμα ΚΔ της σχέσης ΔΡΟΜΕΙΣ, το οποίο θα είναι συστάδων έτσι ώστε να είναι ταξινομημένες οι εγγραφές και να είναι και αποθηκευμένες με τον ίδιο τρόπο, έτσι ώστε να μην διαβάζουμε για κάθε εγγραφή μια σελίδα.
Επίσης, θα δημιουργήσουμε ένα ευρετήριο συστάδων στο γνώρισμα ΚΔ της σχέσης ΤΕΡΜΑΤΙΣΑΝΤΕΣ για τον ίδιο λόγο όπως προηγουμένως.

Ο ΚΔ στους ΔΡΟΜΕΙΣ είναι μοναδικός, επομένως για κάθε 'κδ' θα επιστρέφεται μόνο μια εγγραφή, και επειδή το ευρετήριο βρίσκεται στην μνήμη το κόστος διάσχισης του είναι 0. Ωστόσο επειδή για την προβολή χρειαζόμαστε όλα τα στοιχεία του δρομέα, θα χρειαστεί να ανακτηθεί η αντίστοιχη εγγραφή, οπότε για την συγκεκριμένη πράξη έχουμε κόστος 1 I/O (αφού μια εγγραφή της σχέσης ΔΡΟΜΕΙΣ χωράει σε μια σελίδα).

Ο αλγόριθμος NLJ για κάθε εγγραφή που δέχεται ως είσοδο χρησιμοποιεί το ευρετήριο που υπάρχει στο γνώρισμα ΤΕΡΜΑΤΙΣΑΝΤΕΣ.ΚΔ για να ανακτήσει τις αντίστοιχες εγγραφές της σχέσης ΤΕΡΜΑΤΙΣΑΝΤΕΣ. Δεδομένου ότι έχουμε τους τερματίσαντες μόνο 4 ετών, για κάθε δρομέα θα υπάρχουν το πολύ 4 εγγραφές της σχέσης ΤΕΡΜΑΤΙΣΑΝΤΕΣ.
Οι 4 εγγραφές αυτές χωράνε σε μια σελίδα, οπότε θα διαβαστεί και μια σελίδα λόγω του ότι το ευρετήριο είναι clustered και άρα οι εγγραφές θα βρίσκονται στο ίδιο σημείο. Επομένως, έχουμε κόστος 1 * 1 = 1 I/O

Αρα συνολικά έχουμε κόστος 1 + 1 = 2 I/Os.

Άσκηση 4

Απο τα δεδομένα έχουμε:
T(ΑΚΡΟΑΤΕΣ) = 30000 και μια σελίδα χωράει 10 εγγραφές της σχέσης.

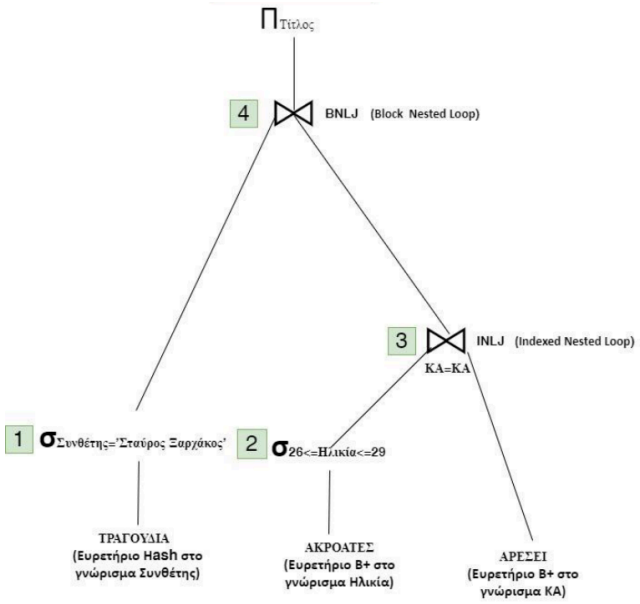
ΑΚΡΟΑΤΕΣ.Ηλικία → [21..60]
T(ΤΡΑΓΟΥΔΙΑ) = 1000 και και μια σελίδα χωράει 5 εγγραφές της σχέσης.
V(ΤΡΑΓΟΥΔΙΑ, Συνθέτης) = 100

T(ΑΡΕΣΕΙ) = 500000 και και μια σελίδα χωράει 50 εγγραφές της σχέσης.

M = 62

Από τα παραπάνω προκύπτουν:
B(ΑΚΡΟΑΤΕΣ) = T(A)/10 = 3000
B(ΤΡΑΓΟΥΔΙΑ) = T(T) / 5 = 200
B(ΑΡΕΣΕΙ) = T(A) / 50 = 10000

Α)



Βήμα 1:

Έστω Χ = σΣυνθέτης='Σταύρος Ξαρχάκος'

Υπάρχουν 1000 τραγούδια και 100 διαφορετικοί συνθέτες. Γνωρίζουμε ότι οι συνθέτες κατανέμονται ομοιόμορφα στα τραγούδια, επομένως

T(X) = $\frac{1000}{100}$ = 10. Στην έξοδο, δηλαδή, θα επιστραφούν 10 εγγραφές. Ακόμα, απαιτούνται

B(X) = $\frac{T(X)}{5}$ = 2 σελίδες για την αποθήκευση των εγγραφών της Χ.

Γνωρίζουμε ότι υπάρχει ένα ευρετήριο κατακερματισμού στο γνώρισμα Συνθέτης. Επομένως, κάθε συνθέτης y θα έχει την δική του καταχώρηση στο ευρετήριο κατακερματισμού (h(key=y)). Η πρόσβαση στον συνθέτη μέσω του ευρετηρίου γίνεται σε O(1) χρόνο και για τον συνθέτη 'Σταύρος Ξαρχάκος' υπάρχουν 2 σελίδες με τα τραγούδια του, άρα η πρόσβαση σε αυτές τις σελίδες θα έχει κόστος 2. Ωστόσο, από την εκφώνηση γνωρίζουμε ότι οι σελίδες του είναι αποθηκευμένες στη μνήμη, οπότε το κόστος αυτό είναι 0. Το ευρετήριο αυτό γνωρίζουμε ότι είναι απλό, δηλαδή non clustered, επομένως υποθέτουμε ότι στη χειρότερη περίπτωση οι 10 εγγραφές θα είναι αποθηκευμένες σε 10 διαφορετικές σελίδες.

Άρα, τελικά cost(X) = 10 I/Os

Βήμα 2:

Έστω ότι Υ = σ26<=Ηλικία<=29

Γνωρίζουμε ότι η έρευνα διεξήχθη σε ακροατές ηλικίας 21 με 60 χρονών. Επομένως υπάρχουν 60 - 21 + 1 = 40 διακριτές τιμές ηλικίας, δηλαδή V(ΑΚΡΟΑΤΕΣ, Ηλικια) = 40

Λόγω ομοιόμορφης κατανομής η πιθανότητα εμφάνισης εγγραφών με ηλικία [26,29] στον πίνακα ΑΚΡΟΑΤΕΣ είναι $\frac{29-26+1}{40} = \frac{1}{10}$. Λόγω ανεξαρτησίας η πιθανότητα αυτή διατηρείται στον πίνακα ΑΚΡΟΑΤΕΣ. Συνεπώς, $\frac{1}{10} \times$ T(ΑΚΡΟΑΤΕΣ) $= \frac{1}{10} \times 30000 = 3000$.

T(Y) = 3000 εγγραφές οι οποίες χωράνε σε B(Y) = $\frac{T(Y)}{10} = 300$ σελίδες.

Γνωρίζουμε ότι υπάρχει ένα B+ ευρετήριο, το οποίο είναι clustered, δηλαδή οι εγγραφές δεν είναι αποθηκευμένες με την ίδια σειρά που είναι στο ευρετήριο. Επομένως για κάθε σελίδα θα κάνουμε μια ξεχωριστή προσπέλαση στον δίσκο. Το κόστος προσπέλασης του B+ ευρετηρίου είναι 0, λόγω του ότι οι σελίδες του ευρετηρίου είναι αποθηκευμένες στην μνήμη.

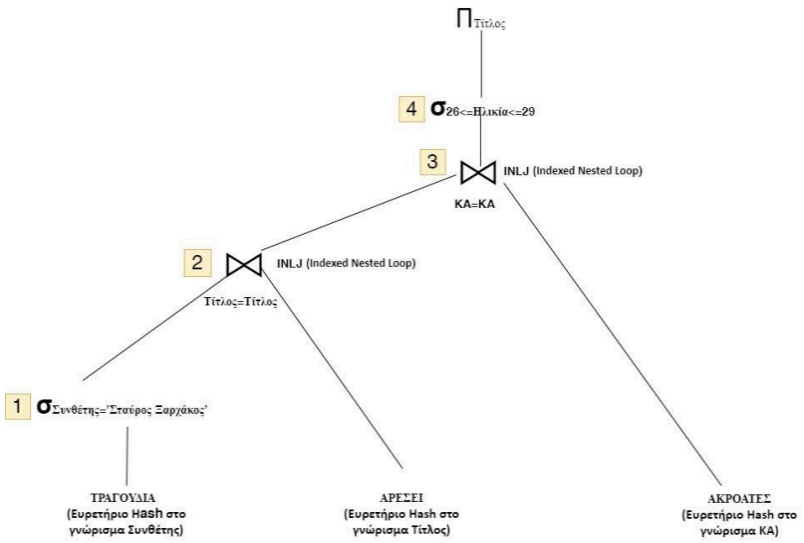
Δηλαδή, cost(Y) = 300 I/Os

Βήμα 3:
Για κάθε μία από τις 3000 εγγραφές που δέχεται ως είσοδο ο αλγόριθμος INLJ χρησιμοποιεί το ευρετήριο που υπάρχει στο γνώρισμα ΑΡΕΣΕΙ.ΚΑ για να ανακτήσει τις αντίστοιχες εγγραφές της σχέσης ΑΡΕΣΕΙ.
Οι τιμές του γνωρίσματος ΚΑ υπάρχουν στην μνήμη διότι βρίσκονται στα φύλλα του ευρετηρίου ΑΡΕΣΕΙ.ΚΑ. Επειδή όμως χρειάζονται οι τιμές του γνωρίσματος Τίτλος πρέπει να ανακτηθούν οι αντίστοιχες εγγραφές.
Επειδή υποθέτουμε ομοιόμορφη κατανομή των τιμών για κάθε εγγραφή που δέχεται ως είσοδο ο INLJ θα επιστρέψει 17 εγγραφές ΑΡΕΣΕΙ:
T(ΑΡΕΣΕΙ)/V(ΑΡΕΣΕΙ, ΚΑ) = 500000/30000 = 17.
Το ευρετήριο στο γνώρισμα ΑΡΕΣΕΙ.ΚΑ είναι ευρετήριο συστάδων, και δεδομένου ότι 17 εγγραφές ΑΡΕΣΕΙ χωράνε σε μία σελίδα, το κόστος είναι 3000 I/O.
Ο αριθμός των εγγραφών στην έξοδο είναι 17*3000 = 51000.

Βήμα 4:
Οι εγγραφές που απαιτούνται για την λειτουργία αυτή βρίσκονται ήδη στην μνήμη από τα προηγούμενα βήματα. Επομένως, το βήμα αυτό μπορεί να εκτελεστεί με μηδενικό κόστος.

Συνολικά το κόστος των 4 βημάτων είναι
10 + 300 + 3000 + 0 = 3310 I/Os

B)



Βήμα 1:
Το βήμα 1 υπολογίζεται όπως και στο ερώτημα Α.
cost(X) = 10 I/Os

Βήμα 2:

Για κάθε μία από τις 10 εγγραφές που δέχεται ως είσοδο ο αλγόριθμος INLJ χρησιμοποιεί το ευρετήριο που υπάρχει στο γνώρισμα ΑΡΕΣΕΙ.Τίτλος για να ανακτήσει τις αντίστοιχες εγγραφές της σχέσης ΑΡΕΣΕΙ.

Γνωρίζουμε ότι ο συνθέτης 'Σταύρος Ξαρχάκος' έχει 10 διαφορετικά τραγούδια. Κάθε ένας από τους 10 τίτλους υποθέτω ότι θα είναι αποθηκευμένος σε διαφορετικό bucket από το ευρετήριο κατακερματισμού λόγω της h(key = Τίτλος).

Για κάθε μια από τις 10 εγγραφές που δέχεται ως είσοδο ο INLJ θα την ταιριάξει με τις εγγραφές που βρίσκονται στο κάθε bucket.
Στη σχέση ΤΡΑΓΟΥΔΙΑ είναι πρωτεύον κλειδί ο Τίτλος, που σημαίνει ότι κάθε τίτλος τραγουδιού είναι και διαφορετικός. Επομένως οι διαφορετικοί τίτλοι είναι όσοι και οι εγγραφές της σχέσης ΤΡΑΓΟΥΔΙΑ, δηλαδή 1000.
Αν υποθέσουμε ότι τα τραγούδια που αρέσουν στους ακροατές κατανέμονται ομοιόμορφα σε κάθε ακροατή τότε για κάθε τίτλο τραγουδιού υπάρχουν
$\frac{\text{T(ΑΡΕΣΕΙ)}}{V(\text{ΑΡΕΣΕΙ, Τίτλος})} = \frac{500000}{1000} = 500$ ακροατές.

Για κάθε τίτλο τραγουδιού, δηλαδή αντιστοιχίζονται 500 ακροατές. Συνεπώς για κάθε μια εγγραφή της σχέσης ΤΡΑΓΟΥΔΙΑ, θα ανακτώνται 500/50 = 10 σελίδες της σχέσης ΑΡΕΣΕΙ.

Επομένως, στην έξοδο της ισοσύνδεσης θα επιστραφούν 10 * 500 = 5000 εγγραφές και το κόστος θα είναι ίσο με

$$\text{cost(INLJ)} = 500 * 10 = 5000 \text{ I/Os}$$

Λόγω του γεγονότος ότι το ευρετήριο στο γνώρισμα ΚΑ είναι συστάδων, διαφορετικά για κάθε μια από τις 500 εγγραφές θα έπρεπε να ψάξουμε στη χειρότερη περίπτωση σε 500 διαφορετικές σελίδες.

Βήμα 3:
Για κάθε μία από τις 5000 εγγραφές που δέχεται ως είσοδο ο αλγόριθμος INLJ χρησιμοποιεί το ευρετήριο που υπάρχει στο γνώρισμα ΑΚΡΟΑΤΕΣ.ΚΑ για να ανακτήσει τις αντίστοιχες εγγραφές της σχέσης ΑΚΡΟΑΤΕΣ.
Δεδομένου ότι το ΚΑ είναι πρωτεύον κλειδί του πίνακα ΑΚΡΟΑΤΕΣ, για κάθε ΚΑ θα ανακτηθεί μόνο μία εγγραφή, δηλαδή θα διαβαστεί μία σελίδα. Επομένως έχουμε 5000*1 = 5000 I/O.

Βήμα 4:

Η επιλογή μπορεί να γίνει στην μνήμη. Στην έξοδο του join υπάρχουν 5000 εγγραφές και υποθέτω ότι η κατανομή των ηλικιών σε αυτές είναι ομοιόμορφη. Η πιθανότητα εμφάνισης εγγραφών με ηλικία [26,29] $\frac{29-26+1}{40} = \frac{1}{10}$ και λόγω ανεξαρτησίας η πιθανότητα αυτή διατηρείται, οπότε από τις 5000 εγγραφές οι 500 θα πληρούν την συνθήκη. Μιας και χρειαζόμαστε μόνο το γνώρισμα Τίτλος (για την προβολή μετέπειτα) και την ηλικία για την επιλογή θα υποθέσω ότι τα γνωρίσματα αυτά μόνο χωράνε ως 5000 εγγραφές στην μνήμη. Οπότε η επιλογή έχει μηδενικό κόστος. Οι 500 εγγραφές χωράνε επίσης στην μνήμη για την προβολή.

Άρα συνολικό κόστος είναι 10 + 5000 + 5000 + 0 = 10010 I/Os.

Γ) Από τον υπολογισμό του κόστους κάθε πλάνου καταλαβαίνουμε ότι καλύτερο είναι το πλάνο Α. Ωστόσο και από την πρώτη όψη μπορούμε να προβλέψουμε ότι το πλάνο Α θα ήταν καλύτερο. Μπορούμε να το δούμε, καθώς στο πρώτο πλάνο γίνεται από νωρίς η επιλογή ηλικίας στους ΑΚΡΟΑΤΕΣ με B+ ευρετήριο (που συνηθίζεται σε range queries), το οποίο είναι και ευρετήριο συστάδων που μειώνει αρκετά το κόστος I/O σε αντίθεση με τα απλά ευρετήρια κατακερματισμού που χρησιμοποιήθηκαν στο πλάνο Β.