

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ

1^η Άσκηση

Λαμπράκης Μιχάλης 2020030077

1)Μετρήσεις

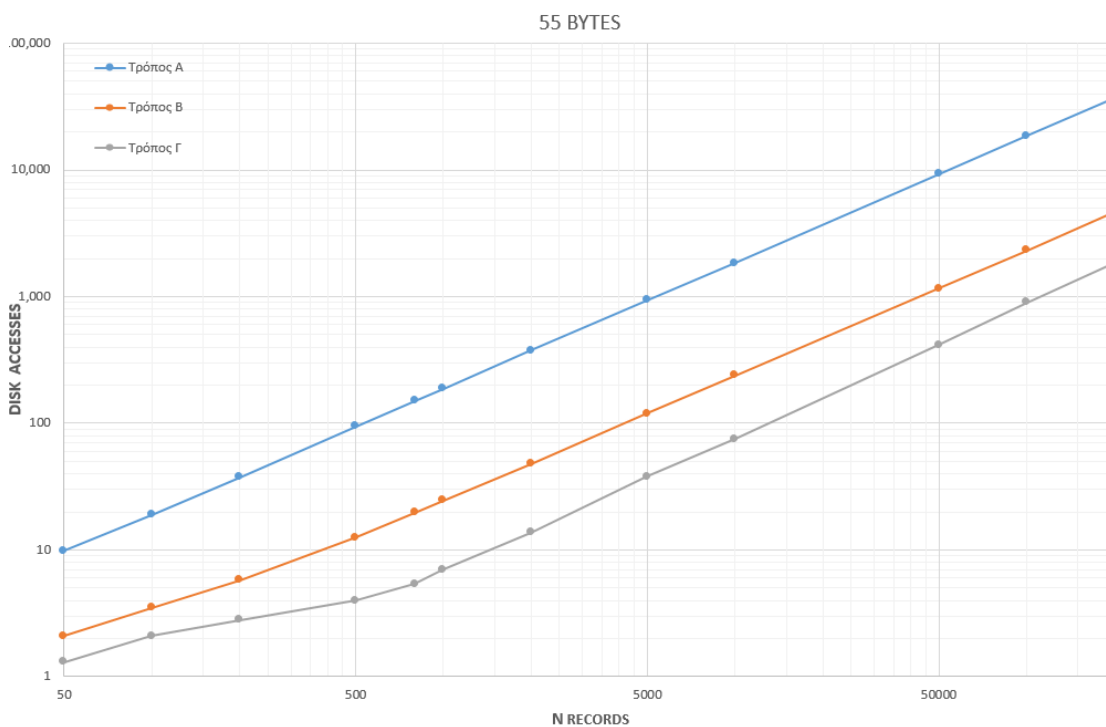
Αρχικά, έχοντας υλοποιήσει σε Java τους 3 ζητούμενους τρόπους οργάνωσης, καταγράφηκαν τα εξής αποτελέσματα των μετρήσεων:

Disk Accesses						
	Τρόπος Α		Τρόπος Β		Τρόπος Γ	
N	55Bytes	27Bytes	55Bytes	27Bytes	55Bytes	27Bytes
50	9.8	5.3	2.1	2.1	1.3	1.3
100	18.9	9.8	3.5	3.5	2.1	2.1
200	37.4	19	5.8	5.7	2.8	2.7
500	94.6	48.1	12.5	12.5	4	4.3
800	150.5	76.2	19.6	19.5	5.4	5.6
1000	186.5	93.6	24.7	24.7	7	7.1
2000	376.6	188.5	47.7	47.8	13.7	12.8
5000	933.4	470.1	118.8	118.1	37.8	34.5
10000	1836.0	948.8	238	235.4	74.8	78.9
50000	9245.7	4623.2	1157.8	1189.5	414.1	429
100000	18483.3	9235.1	2317.4	2354.6	897.3	893.6
200000	37176.5	19132.2	4626.5	4761.4	1853.7	1810.4

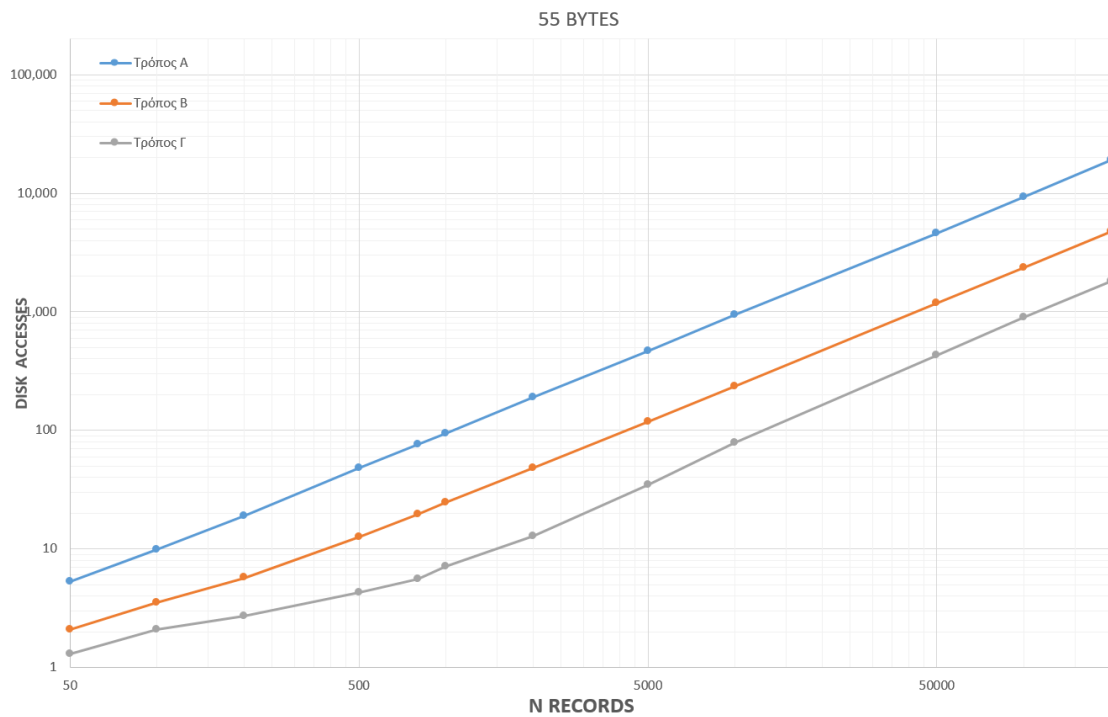
TIME (ns)						
	Τρόπος Α		Τρόπος Β		Τρόπος Γ	
N	55Bytes	27Bytes	55Bytes	27Bytes	55Bytes	27Bytes
50	140128.9	140069.1	215199.7	182704.1	186811.4	164296.6
100	227482.8	246119.2	385661.3	302809.9	278406.9	220897.4
200	386361.3	408199.5	707430.9	584164.5	391374.1	321124.5
500	926217.5	925616.1	1624624.7	1436844.5	551802.4	526921.2
800	1384308.4	1422789.4	2636767	2283146.4	766408.5	684787.2
1000	1731333.3	1702899.4	3180494.9	2871225.6	960553.4	852584.3
2000	3206359.9	3370379.1	6338908.7	5674981.8	1860676	1555836.4
5000	7752456.2	8001796.4	15376993.5	14212851.1	4943824.9	4228832.1
10000	15391838.2	15984017.4	31328664.8	28358875.3	9580004.3	9604498.2
50000	78493817.2	78334432.2	150000289	143664360.8	52445554.3	52415152.3
100000	155984679.2	154157390.9	301606932.1	284943463	112203375.8	108927444.9
200000	301809710.0	318553339.9	589835191.7	584650400.7	240266104.7	220699048.2

2) Διαγράμματα

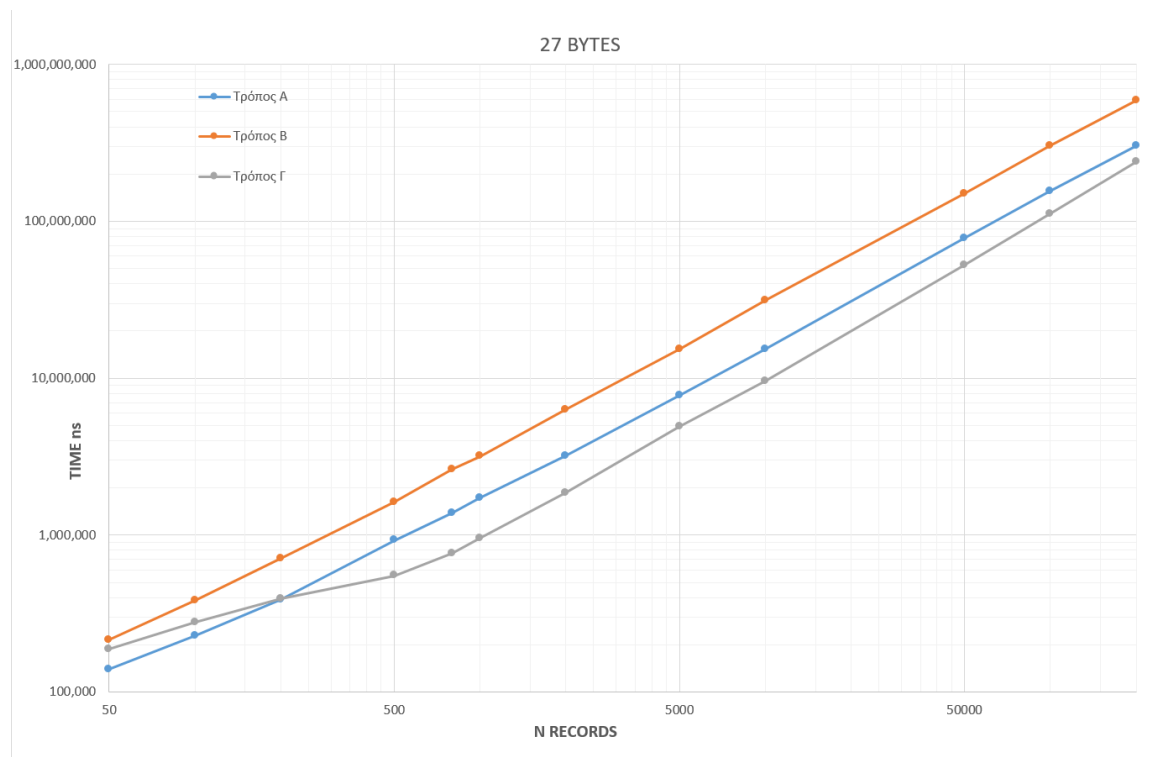
Από τα παραπάνω δεδομένα σχεδιάστηκαν τα διαγράμματα στο Excel χρησιμοποιώντας “Scatter Chart” διαγράμματα και **λογαριθμική κλίμακα** και στους 2 άξονες για να επιτευχθεί μια πιο καθαρή απεικόνιση των αποτελεσμάτων.



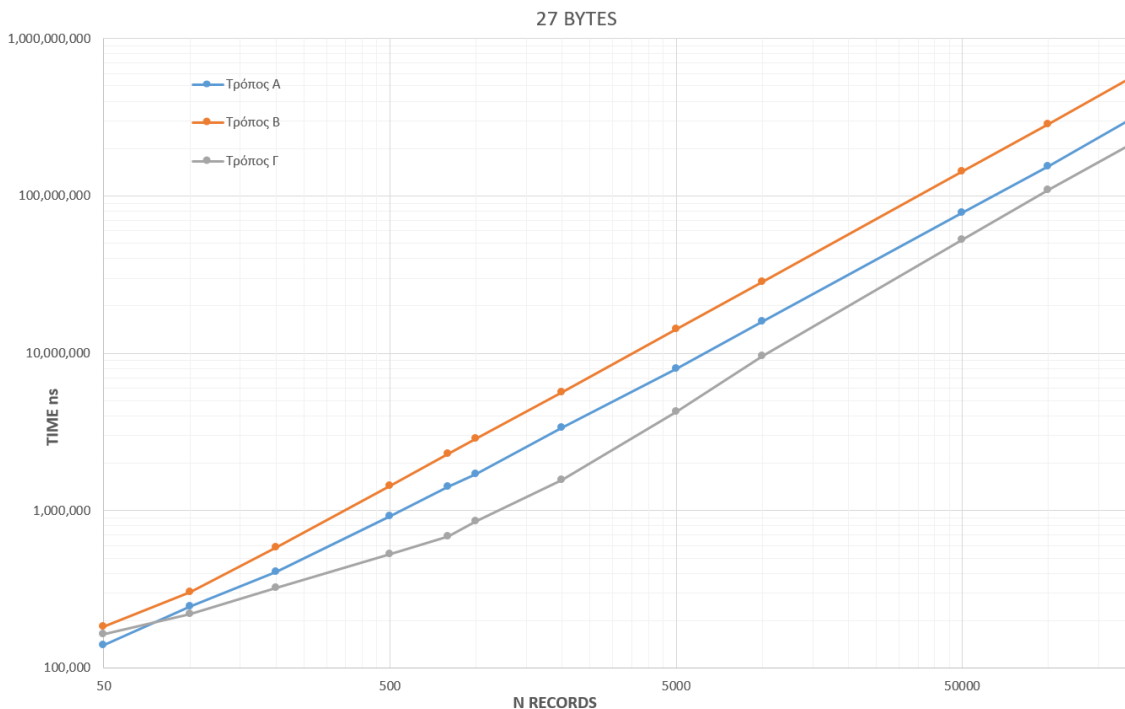
Εικόνα 2.1: Μέσος αριθμός προσβάσεων στο δίσκο για εγγραφές μεγέθους 55 bytes.



Εικόνα 2.2: Μέσος αριθμός προσβάσεων στο δίσκο για εγγραφές μεγέθους 27 bytes.



Εικόνα 2.3: Μέσος χρόνος αναζήτησης (ανά αναζήτηση) για εγγραφές μεγέθους 55 bytes.



Εικόνα 2.4: Μέσος χρόνος αναζήτησης (ανά αναζήτηση) για εγγραφές μεγέθους 27 bytes.

3)Σχόλια & Παρατηρήσεις

Όπως παρατηρούμε σε όλες τις περιπτώσεις η μορφή των καμπυλών είναι γραμμική πράγμα που επιβεβαιώνεται και από τους παραπάνω πίνακες (π.χ ο διπλασιασμός στον αριθμό των εγγραφών προκαλεί και διπλασιασμό στον αριθμό των προσβάσεων στον δίσκο ή στον χρόνο αναζήτησης).

Σε όλες τις περιπτώσεις καθώς αυξάνεται το πλήθος των εγγραφών, αυξάνεται και ο αριθμός των σελίδων των αρχείων οπότε οδηγούμαστε σε περισσότερες προσβάσεις στο δίσκο και σε μεγαλύτερο χρόνο αναζήτησης

Παρατηρούμε ότι στον τρόπο Α όσο μεγαλύτερο είναι το μέγεθος της συμβολοσειράς τόσο πιο πολλές προσβάσεις στον δίσκο πραγματοποιεί το πρόγραμμα και τόσο περισσότερο χρόνο εκτελείται. Στους τρόπου Β και Γ δεν ισχύει αυτό και οι τιμές στα 27 και 55 bytes είναι παρόμοιες και αυτό συμβαίνει διότι η αναζήτηση δεν πραγματοποιείται στο “main” αρχείο αλλά στο αρχείο δεικτοδότησης όπου ο αριθμός των εγγραφών σε κάθε σελίδα δίσκου μένει σταθερός.

Όσον αφορά την σύγκριση των 3 μεθόδων οργάνωσης, παρατηρούμε ότι με τον πρώτο έχουμε πολύ μεγάλο αριθμό προσβάσεων στο δίσκο αφού ψάχνουμε ένα μεγαλύτερο αρχείο και έναν αρκετά μεγάλο χρόνο αναζήτησης. Στον δεύτερο τρόπο η αναζήτηση πραγματοποιείται σε ένα μικρότερο αρχείο (το αρχείο δεικτοδότησης) και αυτό έχει σαν αποτέλεσμα έναν μικρότερο αριθμό προσβάσεων στο δίσκο. Παρόλα αυτά (ίσως σε κάποιο λάθος) ο μέσος χρόνος αναζήτησης είναι λίγο μεγαλύτερος από τον πρώτο τρόπο. Στον τρίτο τρόπο οργάνωσης, η δυαδική αναζήτηση στο ταξινομημένο αρχείο προσφέρει την βέλτιστη απόδοση σε αριθμό προσβάσεων και σε χρόνο εκτέλεσης, καθώς η κάθε επανάληψη του αλγόριθμου μειώνει τον αριθμό των σελίδων που πρέπει να αναζητήσουμε στο μισό.

4)Επεξήγηση κώδικα / κλάσεις και μέθοδοι

Ο Κώδικας περιέχει 6 κλάσεις.

DataClass: Κλάση αναπαράστασης των δεδομένων του βασικού αρχείου. Περιέχει έναν κλειδί (ακέραιος) και μια συμβολοσειρά.

- byteArray: Ενώνει το κλειδί με το String και το επιστρέφει ως byte array.

DataPage: Κλάση αναπαράστασης των στιγμιότυπων του αρχείου δεικτοδότησης. Περιέχει ένα κλειδί και τον αριθμό της σελίδας στο βασικό αρχείο όπου βρίσκεται η εγγραφή και το κλειδί αυτό.

- byteArray: Ενώνει το κλειδί με τον αριθμό σελίδας και το επιστρέφει ως byte array.

FileHandler: Κλάση που διαχειρίζεται οτιδήποτε έχει να κάνει με τα αρχεία (ανάγνωση, εγγραφή και αναζήτηση σε αυτά).

- FileHandler: Δημιουργεί τα αρχεία και αρχικοποιεί τις απαιτούμενες μεταβλητές.
- openFile: Άνοιγμα αρχείων.
- closeFile: Κλείσιμο αρχείων.
- writeBlockFile: Εγγραφή main αρχείου.
- writeBlockIndex: Εγγραφή αρχείου δεικτοδότησης.
- FindRecordA: Αναζήτηση αρχείου με τον τρόπο A.
- FindIndexFilePage: Αναζήτηση στο αρχείο δεικτοδότησης και επιστροφή αριθμού σελίδας που βρίσκεται η εγγραφή στο main αρχείο.
- SearchUsingIndexFile: Αναζήτηση σε συγκεκριμένη σελίδα στο main αρχείο.

- readIndexPage: Συνάρτηση που διαβάζει την ζητούμενη σελίδα από το αρχείο δεικτοδότησης και μετατρέπει τα δεδομένα από bytes σε DataPage objects.
- binarySearch: Υλοποίηση δυαδικής αναζήτησης.

Search: Κλάση η οποία δημιουργεί και αρχικοποιεί DataPage και DataClass objects καθώς και καλεί τον κατάλληλο τρόπο αναζήτησης κάθε φορά.

- Search: Αρχικοποίηση μεταβλητών
- SearchA: Εγγραφή DataClass objects στο αρχείο και πραγματοποίηση αναζήτησης σε αυτό χρησιμοποιώντας τον τρόπο αναζήτησης A.
- SearchB: Εγγραφή DataClass και DataPage objects στα αρχεία και πραγματοποίηση αναζήτησης χρησιμοποιώντας τον τρόπο αναζήτησης B.
- SearchC: Εγγραφή DataClass και DataPage objects στα αρχεία, ταξινόμηση αρχείου δεικτοδότησης και πραγματοποίηση αναζήτησης χρησιμοποιώντας τον τρόπο αναζήτησης Γ.

RandomGenerator: Παράγει τυχαίους ακραίους και τυχαίες συμβολοσειρές.

- getAlphaNumericString: Παραγωγή τυχαίου αλφαριθμητικού.
- getUniqueRandomKey: Παραγωγή ενός array από τυχαίους ακραίους χωρίς να επαναλαμβάνεται κάποιος από αυτούς.
- getRandomKey: Παραγωγή ενός array από τυχαίους (επιτρέπεται η επανάληψη).

Main: Κλάση που δημιουργεί FileHandler και Search objects και τα καλεί κατάλληλα. Τυπώνει στην κονσόλα τον μέσο αριθμό προσβάσεων στον δίσκο για κάθε αριθμό εγγραφών καθώς και τον μέσο χρόνο εκτέλεσης κάθε αναζήτησης.

Ο κύριος όγκος του προγράμματος βρίσκεται στις κλάσεις **FileHandler** και **Search**.

5)Σχετικά με το πρόγραμμα

Το πρόγραμμα δεν απαιτεί ειδικές οδηγίες για να τρέξει. Εκτελεί όλες τις εγγραφές και όλες τις αναζητήσεις με τη σειρά για αυτό και η ολοκλήρωσή του απαιτεί κάποιο χρόνο αφού όταν αυξάνεται το πλήθος των εγγραφών η αναζήτηση σε αυτές είναι πιο χρονοβόρα.

Τυπώνει στην κονσόλα για κάθε ζητούμενο κύκλο αναζητήσεων τον αριθμό προσβάσεων στον δίσκο και τον χρόνο του κύκλου (σε nanosecond). Για να υπολογίσουμε τον μέσο όρο διαιρούμε με το πλήθος των αναζητήσεων (στην περίπτωση μας 1000) και διατηρούμε μόνο το πρώτο δεκαδικό ψηφίο.

Μια παράλογη περίπτωση που δεν μπορώ να βρω που ευθύνεται είναι στον δεύτερο τρόπο αναζήτησης που ενώ ο αριθμός των προσβάσεων στην μνήμη είναι σημαντικά μειωμένος σε σχέση με τον πρώτο τρόπο, ο χρόνος εκτέλεσης του κύκλου αναζητήσεων είναι μεγαλύτερος.

6)Πηγές και εξωτερική βοήθεια.

Το πρόγραμμα γράφτηκε αποκλειστικά από εμένα χωρίς εξωτερική βοήθεια. Χρησιμοποίησα την παράγραφο **‘Χρήσιμα’** που μας δόθηκε στην εκφώνηση και περιείχε πολύ χρήσιμες συναρτήσεις. Όταν αντιμετώπιζα κάποιο πρόβλημα, πέρα από τις διαλέξεις και τα φροντιστήρια, αναζητούσα και ιστοσελίδες στο διαδίκτυο όπως το Stack Overflow, το Geek For Geeks ή ακόμα και την ιστοσελίδα της ίδιας της Oracle.