



Σχολή
Ηλεκτρολόγων
Μηχανικών &
Μηχανικών
Υπολογιστών

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ- Η.Μ.Μ.Υ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ – ΠΛΗ201

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

ΑΤΟΜΙΚΗ ΑΝΑΦΟΡΑ ΥΛΟΠΟΙΗΣΗΣ

Αντώνης Χρυσοφάκης 2015030116

Σκοπός Εργασίας:

Εξοικείωση με πρακτικές συγγραφής κώδικα σε Java ,Με την χρήση αρχείων στο δίσκο και Τεχνικές Αναζήτησης.Επίσης η άσκηση έχει σκοπό την εξοικείωση με δομές δεδομένων και αρχείων που χειρίζονται δεδομένα από διάφορα αρχεία.

Περι Υλοποίησης:

Ο κώδικας για την παρούσα άσκηση γράφτηκε στην γλώσσα Java στο IDE Eclipse, ενώ η ανάλυση και υλοποίηση αυτής καταταμίζεται σε 3 διακριτά υποτομήματα:

- Επεξεργασία στην Κεντρική Μνήμη.
- Επεξεργασία στον δίσκο.
- Σύγκριση μεθόδων και τεκμηρίωση.

Οργάνωση Κώδικα και Documentation:

Ο πηγαίος κώδικας βρίσκεται οργανωμένος σε 3 πακέτα (packages) τα περιεχόμενα των οποίων περιγράφονται αναλυτικά στον παρακάτω πίνακα.

Packages:	Use:
<i>Files</i>	Δημιουργία όλων των αρχείων που θα χρησιμοποιήσουμε(MakeFile,MakeIndexFile,MakeSortedIndexFile).
<i>methods</i>	Εδώ βρίσκονται όλοι οι μέθοδοι αναζήτησης ενός κλειδιού(MethodA,MethodB,MethodC).

<i>pck</i>	Κυρίως κλάση (App.java), δημιουργία τυχαίων Strings(RandomString), τέλος εδώ βρίσκονται οι κλάσεις DataClass, DataPage.
------------	---

Υλοποίηση:

Αρχικά δημιουργήσαμε N τυχαία μοναδικά κλειδιά στη μνήμη με τιμές κλειδιών από 1 έως 2xN. Για κάθε κλειδί δημιουργήσαμε μία τυχαία συμβολοσειρά ASCII χαρακτήρων για το πεδίο δεδομένων (55 ή 27 bytes για κάθε περίπτωση) με την χρήση της κλάσης RandomString. Για κάθε κλειδί και συμβολοσειρά, δημιουργήσαμε ένα στιγμιότυπο της κλάσης DataClass. Οπότε έχουμε ένα array με N στιγμιότυπα το οποίο θέλουμε να αποθηκεύσουμε στον δίσκο μας. Επίσης να σημειωθεί ότι οι τιμές που παίρνει το N είναι οι εξής:

- 50, 100, 200, 500, 800, 1.000, 2.000, 5.000, 10.000, 50.000, 100.000, 200.000

Δημιουργία αρχείων:

Στην συνέχεια καλούμε την μέθοδο processArray η οποία βρίσκεται μέσα στην κλάση **MakeFile** μέσω της οποίας δημιουργούμε το κύριο αρχείο μας (κάνοντας διαδοχικές εγγραφές στο δίσκο σελίδων δίσκου) στο οποίο βρίσκονται τα στιγμιότυπα της κλάσης DataClass.

Έπειτα μεταφερόμαστε στην κλάση **MakeIndexFile** όπου μέσω της μεθόδου της processArray δημιουργούμε το Index αρχείο μας (κάνοντας διαδοχικές εγγραφές στο δίσκο σελίδων δίσκου) στο οποίο βρίσκονται τα στιγμιότυπα της κλάσης DataPage.

Τέλος μέσω της κλάσης **MakeSortedIndexFile** και της μεθόδου της processArray, αφού έχουμε ταξινομήσει τα στιγμιότυπα της κλάσης DataPage στη μνήμη, δημιουργούμε το SortedIndex αρχείο μας (κάνοντας διαδοχικές εγγραφές στο δίσκο σελίδων δίσκου) στο οποίο βρίσκονται τα ταξινομημένα στιγμιότυπα της κλάσης DataPage.

Μέθοδοι Αναζήτησης:

Αφού φτιαξαμε τα τρία παραπάνω αρχεία σκοπός μας είναι να δημιουργήσουμε 1000 τυχαία κλειδιά για κάθε N στιγμιότυπων και να χρησιμοποιήσουμε τρεις διαφορετικές μεθόδους αναζήτησης αυτών των τυχαίων κλειδιών μέσα στα αρχεία μας.

- **MethodA:** Σε αυτή την κλάση εκτελείται η μέθοδος searchKey η οποία ουσιαστικά διαβάζει το κύριο αρχείο μας σελίδα-σελίδα και μετατρέπει τα bytes του σε στιγμιότυπα της DataClass. Στη συνέχεια για κάθε σελίδα συγκρίνουμε τα κλειδιά που βρίσκονται μέσα σε αυτήν με το κλειδί που αναζητούμε και αν ταυτίζονται σταματάμε την εκτέλεση και επιστρέφουμε στην main μας τον αριθμό των προσβάσεων που

χρειάστηκαν μέχρι να βρούμε το κλειδί, ή μέχρι να διαβάσουμε όλο το αρχείο μας και να διαπιστώσουμε ότι δεν υπάρχει μέσα σε αυτό.

- MethodB:** Στη συνέχεια μέσω της κλάσης αυτής επικαλούμαστε τη μέθοδο `searchKey` η οποία διαβάζει αυτή τη φορά το `Index` αρχείο μας σελίδα-σελίδα και μετατρέπει τα bytes του σε στιγμιότυπα της `PageData`. Για κάθε σελίδα συγκρίνουμε τα κλειδιά που βρίσκονται μέσα σε αυτήν με το κλειδί που αναζητούμε και αν το βρούμε σταματάμε την εκτέλεση και αφού ξέρουμε σε ποια σελίδα του κύριου αρχείου μας βρίσκεται το συγκεκριμένο κλειδί (εφόσον είναι αποθηκευμένα ως ζεύγη κλειδί-θέση στην `PageData`) διαβάζουμε την συγκεκριμένη σελίδα από το κύριο αρχείο μας και βρίσκουμε και το αντίστοιχο `String` που συνοδεύει το κλειδί μας. Τέλος επιστρέφουμε των αριθμό των προσβάσεων στον δίσκο που χρειάστηκαν.
- MethodC:** Για την τελευταία μέθοδο αναζήτησης εισερχόμαστε στην συγκεκριμένη κλάση όπου με την βοήθεια της μεθόδου `binarySearch` διαβάζουμε το `SortedIndex` αρχείο που δημιουργήσαμε μετατρέποντας τα bytes του σε στιγμιότυπα της κλάσης `PageData`. Αυτή τη φορά αντί να διαβάσουμε όλο το αρχείο μας χρησιμοποιούμε την τεχνική της Δυαδικής αναζήτησης και σε περίπτωση που βρούμε το κλειδί που αναζητάμε σταματάμε την εκτέλεση και όπως και στην `MethodB` ξέρουμε σε ποια σελίδα του κύριου αρχείου μας βρίσκεται το συγκεκριμένο κλειδί την οποία θα διαβάσουμε για να βρούμε το αντίστοιχο `String` το οποίο και μετατρέπουμε από bytes σε `DataClass` στιγμιότυπο. Ομοίως με τις άλλες μεθόδους επιστρέφουμε των αριθμό των προσβάσεων στον δίσκο που χρειάστηκαν για να βρούμε ή και όχι το κλειδί που ψάχναμε.

ΠΕΙΡΑΜΑ:

Αφού δημιουργήσουμε τα αρχεία μας με N στοιχεία κάθε φορά, κάνουμε 1000 αναζητήσεις με τυχαία κλειδιά από 1 έως $2 \times N$ (για N από 50 έως 1000, τα κλειδιά αναζήτησης μπορεί να επαναλαμβάνονται. Για $N > 1000$, τα κλειδιά πρέπει να είναι μοναδικά). Τα αποτελέσματα που παίρνουμε φαίνονται στους παρακάτω πίνακες και εικόνες.

ΠΙΝΑΚΑΣ Α

	ΤΡΟΠΟΣ Α		ΤΡΟΠΟΣ Β		ΤΡΟΠΟΣ Γ	
N	Αριθμός προσβάσεων ανά αναζήτηση . Μεγεθος δεδομένων ν 55 bytes	Αριθμός προσβάσεων ανά αναζήτηση . Μεγεθος δεδομένων ν 27 byte	Αριθμός προσβάσεων ανά αναζήτηση . Μεγεθος δεδομένων ν 55 bytes	Αριθμός προσβάσεων ανά αναζήτηση . Μεγεθος δεδομένων ν 27 bytes	Αριθμός προσβάσεων ανά αναζήτηση . Μεγεθος δεδομένων ν 55 bytes	Αριθμός προσβάσεων ανά αναζήτηση . Μεγεθος δεδομένων ν 27 bytes
50	9.9	5.3	2.2	2.2	2.2	2.2

100	18.8	9.9	3.5	3.5	2.5	2.5
200	39.1	19.0	6.0	5.8	3.2	3.2
500	95.2	47.8	12.8	12.8	4.3	4.3
800	150.2	74.8	19.5	19.4	4.9	4.9
1000	187.9	93.2	24.6	24.4	5.1	5.1
2000	375.5	189.9	47.9	48.4	6.0	6.1
5000	940.8	472.0	118.7	119.0	7.5	7.3
10000	1872.8	930.4	235.1	233.5	8.4	8.4
50000	9164.1	4644.6	1146.5	1162.1	10.7	10.7
100000	18752.7	9428.1	2344.8	2357.7	11.7	11.7
200000	37541.8	18485.3	4693.4	4622.0	12.7	12.7

ΠΙΝΑΚΑΣ Β

	ΤΡΟΠΟΣ Α		ΤΡΟΠΟΣ Β		ΤΡΟΠΟΣ Γ	
N	Χρόνος ανά αναζήτηση . Μεγεθος δεδομένων ν 55 bytes	Χρόνος ανά αναζήτηση . Μεγεθος δεδομένων 27 byte	Χρόνος ανά αναζήτηση . Μεγεθος δεδομένων ν 55 bytes	Χρόνος ανά αναζήτηση . Μεγεθος δεδομένων ν 27 bytes	Χρόνος ανά αναζήτηση . Μεγεθος δεδομένων ν 55 bytes	Χρόνος ανά αναζήτηση . Μεγεθος δεδομένων ν 27 bytes
50	370.6	219.8	310.1	305.4	147.2	116.0
100	316.0	557.6	250.2	254.0	212.4	334.5
200	747.0	413.7	304.4	260.8	153.7	139.1
500	1215.8	843.8	498.5	490.5	140.0	156.0
800	2022.0	1438.6	317.7	342.6	137.6	180.4
1000	1955.7	1256.1	396.2	453.9	293.5	133.0
2000	6498.8	2508.5	873.6	866.7	445.5	178.2
5000	34352.9	4916.5	1878.7	1266.7	480.3	258.5

10000	58531.6	11906.1	3854.7	3166.9	729.1	212.3
50000	145027.6	53469.4	7704.5	13615.5	1056.4	312.2
100000	654636.3	129489.7	28590.4	27330.5	1004.4	337.1
200000	1464810.2	320653.4	56764.3	64237.2	1078.6	386.4

Παρακάτω φαίνονται τα αποτελέσματα του προγράμματος για Μέγεθος Δεδομένων=55bytes:

```

Number of N is: 50 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 9.9 and MO of nanoseconds is: 370.6
Number of N is: 50 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 2.2 and MO of nanoseconds is: 310.1
Number of N is: 50 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 2.2 and MO of nanoseconds is: 147.2
Number of N is: 100 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 18.8 and MO of nanoseconds is: 316.0
Number of N is: 100 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 3.5 and MO of nanoseconds is: 250.2
Number of N is: 100 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 2.5 and MO of nanoseconds is: 212.4
Number of N is: 200 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 39.1 and MO of nanoseconds is: 747.0
Number of N is: 200 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 6.0 and MO of nanoseconds is: 304.4
Number of N is: 200 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 3.2 and MO of nanoseconds is: 153.7
Number of N is: 500 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 95.2 and MO of nanoseconds is: 1215.8
Number of N is: 500 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 12.8 and MO of nanoseconds is: 498.5
Number of N is: 500 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 4.3 and MO of nanoseconds is: 140.0
Number of N is: 800 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 150.2 and MO of nanoseconds is: 2022.0
Number of N is: 800 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 19.5 and MO of nanoseconds is: 317.7
Number of N is: 800 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 4.9 and MO of nanoseconds is: 137.6
Number of N is: 1000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 187.9 and MO of nanoseconds is: 1955.7
Number of N is: 1000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 24.6 and MO of nanoseconds is: 396.2
Number of N is: 1000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 5.1 and MO of nanoseconds is: 293.5
Number of N is: 2000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 375.5 and MO of nanoseconds is: 6498.8
Number of N is: 2000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 47.9 and MO of nanoseconds is: 873.6
Number of N is: 2000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 6.0 and MO of nanoseconds is: 445.5
Number of N is: 5000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 940.8 and MO of nanoseconds is: 34352.9
Number of N is: 5000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 118.7 and MO of nanoseconds is: 1878.7
Number of N is: 5000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 7.5 and MO of nanoseconds is: 480.3
Number of N is: 10000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 1872.8 and MO of nanoseconds is: 58531.6
Number of N is: 10000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 235.1 and MO of nanoseconds is: 3854.7
Number of N is: 10000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 8.4 and MO of nanoseconds is: 729.1
Number of N is: 50000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 9164.1 and MO of nanoseconds is: 145027.6
Number of N is: 50000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 1146.5 and MO of nanoseconds is: 7704.5
Number of N is: 50000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 10.7 and MO of nanoseconds is: 1056.4
Number of N is: 100000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 18752.7 and MO of nanoseconds is: 654636.3
Number of N is: 100000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 2344.8 and MO of nanoseconds is: 28590.4
Number of N is: 100000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 11.7 and MO of nanoseconds is: 1004.4
Number of N is: 200000 Field Size(in bytes) is: 55 MO of accesses in MethodA is: 37541.8 and MO of nanoseconds is: 1464810.2
Number of N is: 200000 Field Size(in bytes) is: 55 MO of accesses in MethodB is: 4693.4 and MO of nanoseconds is: 56764.3
Number of N is: 200000 Field Size(in bytes) is: 55 MO of accesses in MethodC is: 12.7 and MO of nanoseconds is: 1078.6

```

Παρακάτω φαίνονται τα αποτελέσματα του προγράμματος για Μέγεθος Δεδομένων=27 bytes:

Number of N is:	50	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	5.3	and MO of nanoseconds is:	219.8
Number of N is:	50	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	2.2	and MO of nanoseconds is:	305.4
Number of N is:	50	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	2.2	and MO of nanoseconds is:	116.0
Number of N is:	100	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	9.9	and MO of nanoseconds is:	557.6
Number of N is:	100	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	3.5	and MO of nanoseconds is:	254.0
Number of N is:	100	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	2.5	and MO of nanoseconds is:	334.5
Number of N is:	200	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	19.0	and MO of nanoseconds is:	413.7
Number of N is:	200	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	5.8	and MO of nanoseconds is:	260.8
Number of N is:	200	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	3.2	and MO of nanoseconds is:	139.1
Number of N is:	500	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	47.8	and MO of nanoseconds is:	843.8
Number of N is:	500	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	12.8	and MO of nanoseconds is:	490.5
Number of N is:	500	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	4.3	and MO of nanoseconds is:	156.0
Number of N is:	800	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	74.8	and MO of nanoseconds is:	1438.6
Number of N is:	800	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	19.4	and MO of nanoseconds is:	342.6
Number of N is:	800	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	4.9	and MO of nanoseconds is:	180.4
Number of N is:	1000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	93.2	and MO of nanoseconds is:	1256.1
Number of N is:	1000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	24.4	and MO of nanoseconds is:	453.9
Number of N is:	1000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	5.1	and MO of nanoseconds is:	133.0
Number of N is:	2000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	189.8	and MO of nanoseconds is:	2508.5
Number of N is:	2000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	48.4	and MO of nanoseconds is:	866.7
Number of N is:	2000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	6.1	and MO of nanoseconds is:	178.2
Number of N is:	5000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	472.0	and MO of nanoseconds is:	4916.5
Number of N is:	5000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	119.0	and MO of nanoseconds is:	1266.7
Number of N is:	5000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	7.3	and MO of nanoseconds is:	258.5
Number of N is:	10000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	930.4	and MO of nanoseconds is:	11906.1
Number of N is:	10000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	233.5	and MO of nanoseconds is:	3166.9
Number of N is:	10000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	8.4	and MO of nanoseconds is:	212.3
Number of N is:	50000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	4644.6	and MO of nanoseconds is:	53469.4
Number of N is:	50000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	1162.1	and MO of nanoseconds is:	13615.5
Number of N is:	50000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	10.7	and MO of nanoseconds is:	312.2
Number of N is:	100000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	9428.1	and MO of nanoseconds is:	129489.7
Number of N is:	100000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	2357.7	and MO of nanoseconds is:	27330.5
Number of N is:	100000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	11.7	and MO of nanoseconds is:	337.1
Number of N is:	200000	Field Size(in bytes) is:	27	MO of accesses in MethodA is:	18485.3	and MO of nanoseconds is:	320653.4
Number of N is:	200000	Field Size(in bytes) is:	27	MO of accesses in MethodB is:	4622.0	and MO of nanoseconds is:	64237.2
Number of N is:	200000	Field Size(in bytes) is:	27	MO of accesses in MethodC is:	12.7	and MO of nanoseconds is:	386.4

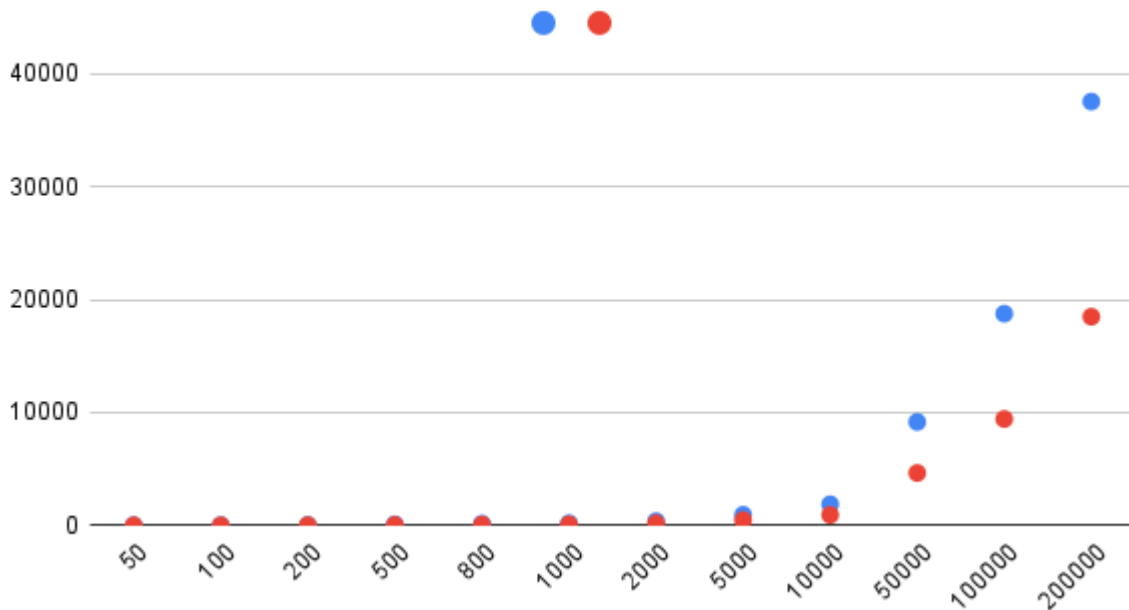
Διαγράμματα-Σχολιασμός Αποτελεσμάτων:

Σε όλα τα διαγράμματα το **μπλε** χρώμα αντιπροσωπεύει το **Μέγεθος δεδομένων 55 bytes**, ενώ το **κόκκινο** το **Μέγεθος δεδομένων 27 bytes**:

ΤΡΟΠΟΣ Α:

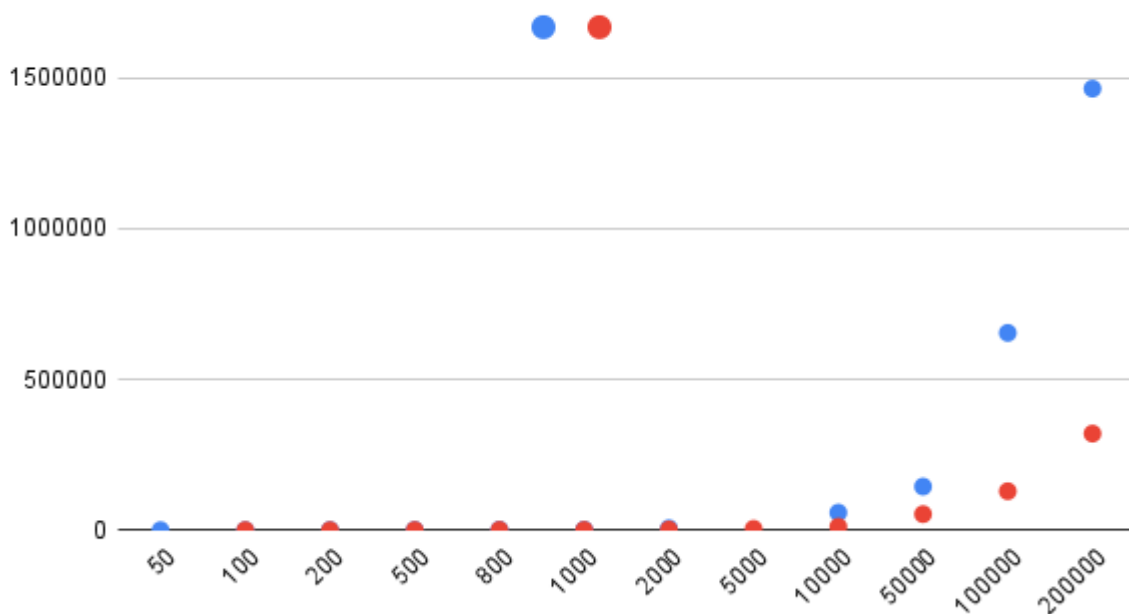
Στο **διάγραμμα 1.1** φαίνεται ο Μέσος όρος Προσβάσεων στο δίσκο σε συνάρτηση με το N, ενώ στο **διάγραμμα 1.2** φαίνεται ο Μέσος χρόνος ανα αναζήτηση σε συνάρτηση με το N.

ΤΡΟΠΟΣ Α(ΜΟ ΠΡΟΣΒΑΣΕΩΝ)



Διαγράμμα 1.1

ΤΡΟΠΟΣ Α(Μέσος Χρόνος Εκτέλεσης)



ΔΙΑΓΡΑΜΜΑ 1.2

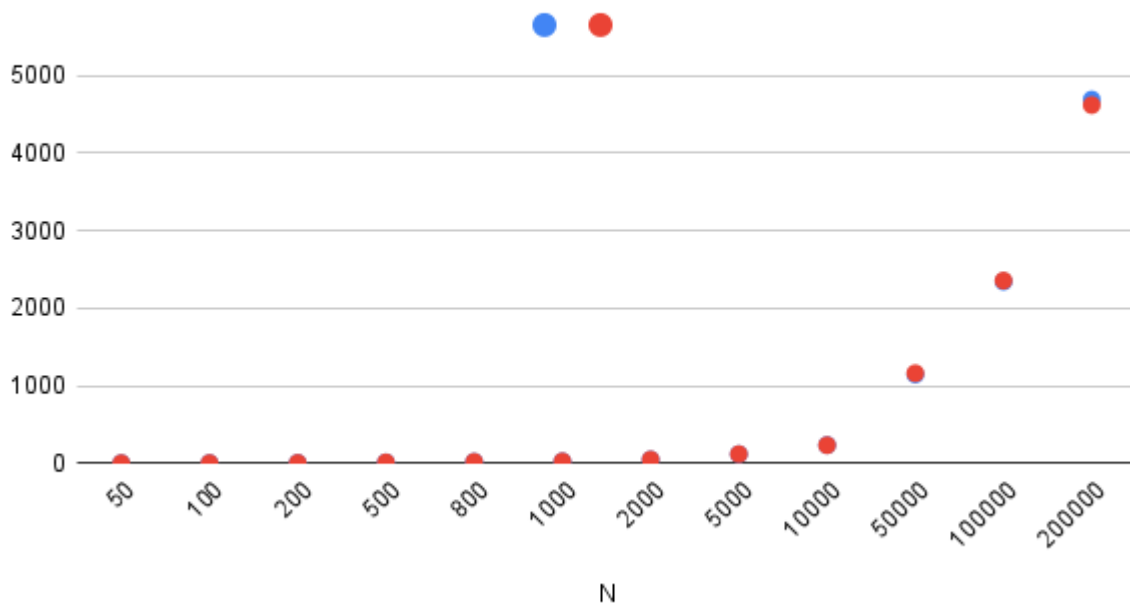
Είναι φανερό ότι οι μορφές των καμπυλών μας είναι εκθετικές. Αυτό δικαιολογείται αφού όσο μεγαλύτερο το N τόσο μεγαλύτερο θα είναι το κύριο αρχείο μας που συνεπάγεται ότι με την συγκεκριμένη μέθοδο αναζήτησης θα έχουμε περισσότερες προσβάσεις για την εύρεση ή μη του κλειδιού μας (άρα και παραπάνω χρόνος). Όταν το Μέγεθος δεδομένων είναι 27 bytes έχουμε τις μισές προσβάσεις όπως είναι λογικό αφού σε κάθε σελίδα χωράνε τα διπλά

στιγμιότυπα σε σχέση με τα 55 bytes, και επομένως έχουμε τις μισές σελίδες άρα και τις μισές προσβάσεις στο δίσκο.

ΤΡΟΠΟΣ Β:

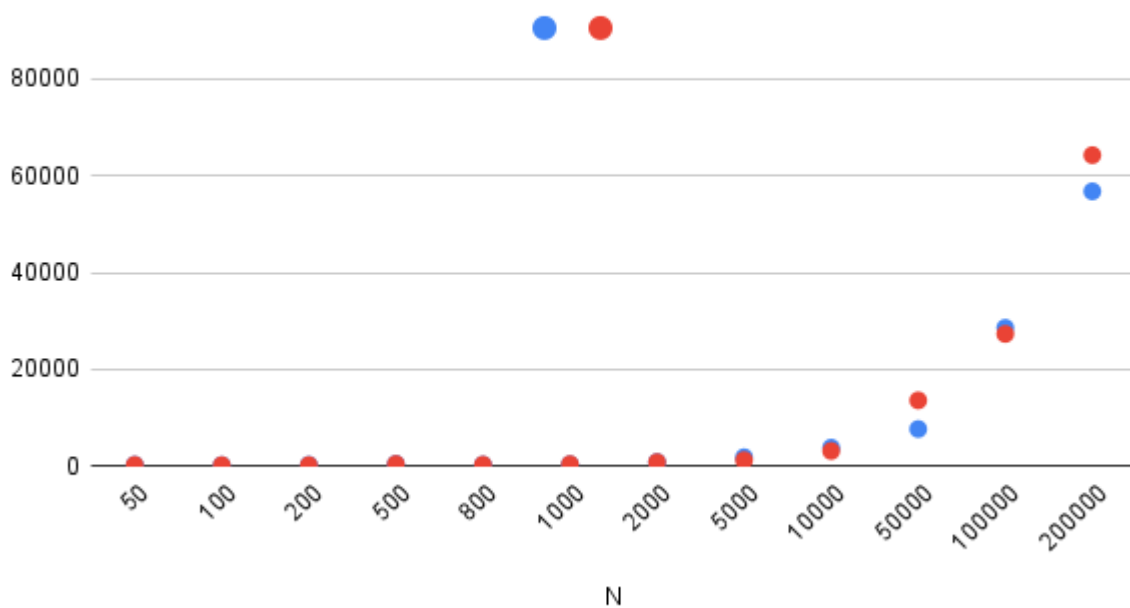
Στο **διάγραμμα 2.1** φαίνεται ο Μέσος όρος Προσβάσεων στο δίσκο σε συνάρτηση με το N, ενώ στο **διάγραμμα 2.2** φαίνεται ο Μέσος χρόνος ανά αναζήτηση σε συνάρτηση με το N.

ΤΡΟΠΟΣ Β (ΜΟ ΠΡΟΣΒΑΣΕΩΝ)



Διάγραμμα 2.1

ΤΡΟΠΟΣ Β (Μέσος Χρόνος ανά αναζήτηση)



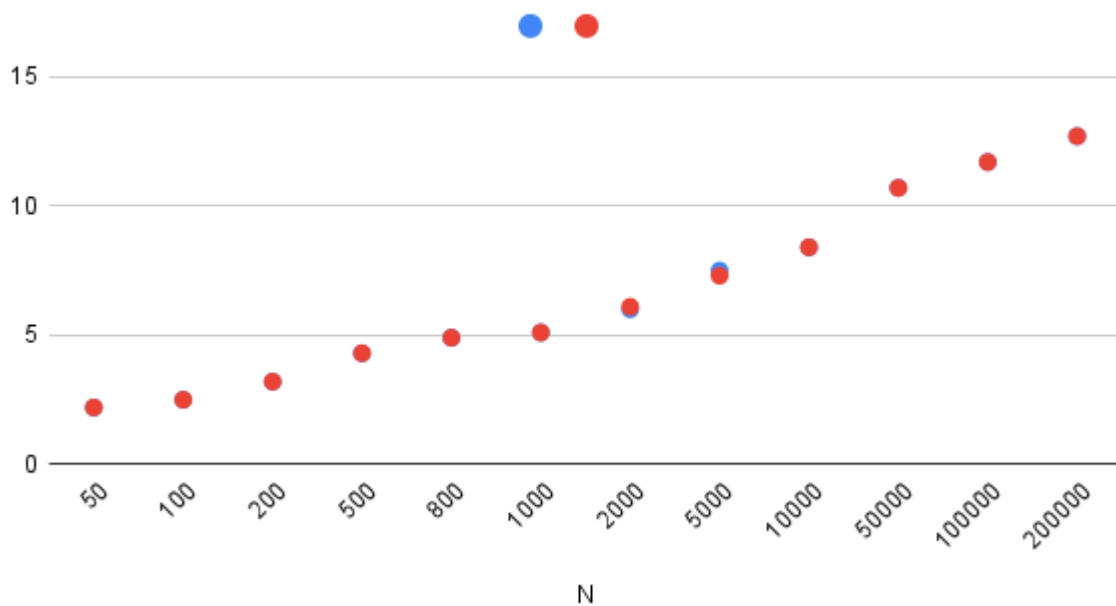
Διάγραμμα 2.2

Χρησιμοποιώντας αυτόν τον τρόπο αναζήτησης βλέπουμε ότι οι μορφές των καμπυλών μας είναι και πάλι εκθετικές αλλά με πολύ λιγότερες προσβάσεις(συνεπώς και χρόνο).Αυτο συμβαίνει διότι όσο μεγαλύτερο το N τόσο μεγαλύτερο θα είναι το αρχείο δεικτοδότησης.Επομένως εφόσον αυτή η μέθοδος προσπελαύνει απο την αρχή το αρχείο δεικτοδότησης έως ότου τελος θα έχουμε ένα μεγάλο αριθμό προσβάσεων για την εύρεση ή μη του κλειδιού μας (άρα και αρκετό χρόνο).Όταν το Μέγεθος δεδομένων είναι 27 bytes έχουμε σχεδόν τις ίδιες προσβάσεις(στο μέσο χρόνο έχουμε αμεληταίες διαφορές) αφού στο Index File δεν μας επηρεάζει το μέγεθος του String(εφόσον δεν το χρησιμοποιούμε).

ΤΡΟΠΟΣ Γ:

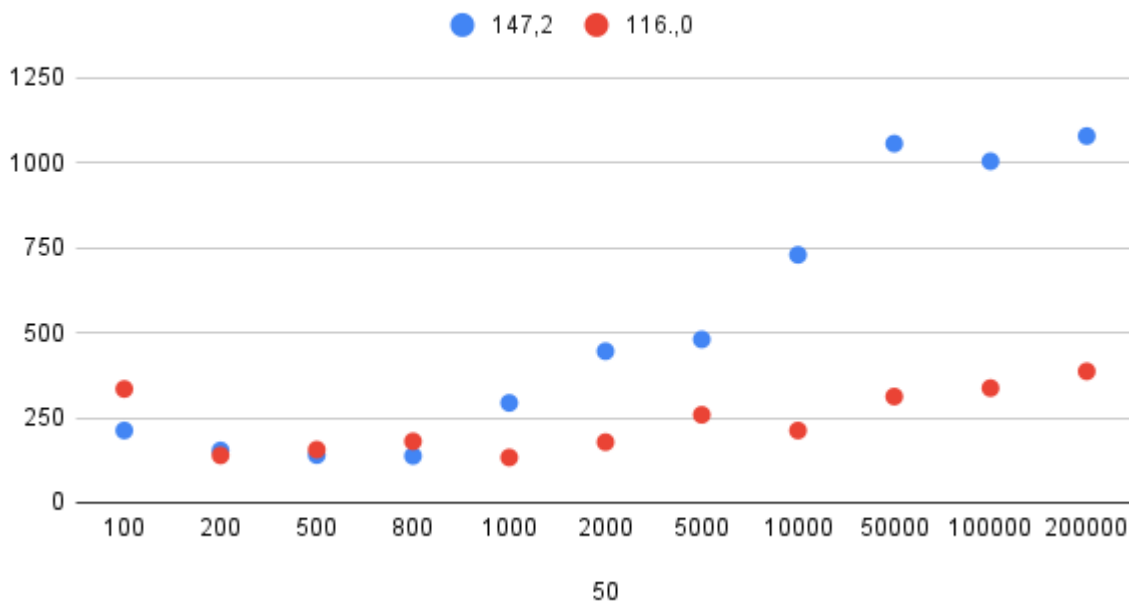
Στο **διάγραμμα 3.1** φαίνεται ο Μέσος όρος Προσβάσεων στο δίσκο σε συνάρτηση με το N,ενώ στο **διάγραμμα 3.2** φαίνεται ο Μέσος χρόνος ανα αναζήτηση σε συνάρτηση με το N.

ΤΡΟΠΟΣ Γ(ΜΟ ΠΡΟΣΒΑΣΕΩΝ)



Διάγραμμα 3,1

ΤΡΟΠΟ Γ(Μέσος Χρόνος Ανα Αναζήτηση)



Διάγραμμα 3.2

Τέλος με τον τελευταίο τρόπο αναζήτησης που χρησιμοποιούμε παρατηρούμε ότι οι μορφές των καμπυλών μας τείνουν να γίνουν γραμμικές με τις λιγότερες προσβάσεις από όλους τους προηγούμενους τρόπους που δοκιμάσαμε(συνεπώς και με τον λιγότερο χρόνο).Αυτο συμβαίνει διότι όσο μεγάλο και αν είναι το N , με τη χρήση της δυαδικής αναζήτησης δεν χρειάζεται να διαβάζουμε όλο το ταξινομημένο αρχείο δεικτοδότησης αλλά μόνο κάποιες σελίδες του.Όταν το Μέγεθος δεδομένων είναι 27 bytes έχουμε σχεδόν τις ίδιες προσβάσεις αφού όπως είπαμε και παραπάνω στο Index File δεν μας επηρεάζει το μέγεθος του String(εφόσον δεν το χρησιμοποιούμε).Στο διάγραμμα του μέσου χρόνου ανα αναζήτηση έχουμε κάποιες διαφορές στα nanoseconds οι οποίες μάλλον οφείλονται στο γεγονός ότι εάν βρω το κλειδί που ψάχνω για να αποκωδικοποιήσω τα 55 bytes έτσι ώστε να πάρω την String πληροφορία μου θέλω παραπάνω χρόνο απ ότι τα 27 bytes.

Συμπεράσματα:

Σκοπός του παρόντος εδαφίου αποτελεί η σύγκριση των τεχνικών αναζήτησης που υλοποιήσαμε. Βασιζόμενοι στους πίνακες σύγκρισης προσβάσεων αλλά και στα διαγράμματα μπορεί εύκολα να εξαχθεί το γεγονός ότι,η Δυαδική Αναζήτηση που πραγματοποιήσαμε στην MethodC είναι τάξεις μεγέθους αποδοτικότερη σε θέμα συγκρίσεων (άρα και ταχύτητας και υπολογιστικού φόρτου). Το παραπάνω γεγονός έγκειται στη φύση των

αναζητήσεων καθότι η αναζήτηση του ερωτήματος A(MethodA) πρέπει να ψάξει από την αρχή το κύριο αρχείο μας έως ότου βρει το ζητούμενο κλειδί και στην χειρότερη περίπτωση προσπελαύνοντας όλα τα στοιχεία του αρχείου(μη εύρεση κλειδιού).Μπορεί η αναζήτηση του ερωτήματος B(MethodB) να είναι πιο αποδοτική από την A αλλά αυτό δεν αρκεί για να πούμε ότι είναι μια αποτελεσματική μέθοδος αναζήτησης, διότι και αυτή πρέπει να ψάξει από την αρχή το αρχείο δεικτοδότησης(το οποίο είναι αισθητά πιο μικρό από το κύριο αρχείο μας) έως ότου βρει το ζητούμενο κλειδί και στην χειρότερη περίπτωση προσπελαύνοντας όλα τα στοιχεία του αρχείου(μη εύρεση κλειδιού).

Παραπομπές – Πηγές:

- 1. Φροντιστήριο Μαθήματος.**
- 2. Σελίδα Μαθήματος eclass.**
- 3. <https://www.geeksforgeeks.org/generate-random-string-of-given-size-in-java/>**