

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩ ΚΑΙ ΑΛΓΟΡΙΘΜΩΝ

Αναφορά 1ης εργαστηριακής άσκησης

Χαϊντούτη Μαρία || AM 2020030129

Η main μέθοδος βρίσκεται στο πακέτο Main και αποτελείται από τις μεθόδους methodA, methodB, methodC, οι οποίες δημιουργούνται έξω από αυτή και η main απλώς τις καλεί μετά την δημιουργία των απαραίτητων αρχείων.

Μέθοδος A:

Η μέθοδος A αποτελείται από τις κλάσεις DataClass, WritingAndReading και DataPage.

DataClass:

- **DataClass():** Αυτή η μέθοδος είναι ο constructor της κλάσης DataClass και χρησιμοποιείται για την δημιουργία αντικειμένων DataClass με ένα κλειδί(key) και ένα αλφαριθμητικό(data).
- Ακολουθούν οι getters ,οι οποίοι επιστρέφουν τα αντικείμενα της DataClass και οι setters, οι οποίοι χρησιμοποιούνται για τον ορισμό των αντικειμένων.
- **Public static DataClass[] byteToDataArray():** Αυτή η μέθοδος χρησιμοποιείται για την μετατροπή ενός πίνακα αντικειμένων DataClass σε byte array.
- **Public static DataClass[] byteToDataArray():** αντίστοιχα αυτή η μέθοδος είναι υπεύθυνη για μετατροπή ενός byte[] σε έναν πίνακα DataClass.
- **Public static DataClass searchInDataFile():** χρησιμοποιείται για την αναζήτηση ενός στοιχείου με βάση το κλειδί σε ένα αρχείο δεδομένων.

WritingAndReading:

- **WritingAndReading(File f):** Αυτή η μέθοδος είναι ο constructor της κλάσης και δέχεται ένα αντικείμενο τύπου File ως όρισμα. Αρχικοποιεί τα πεδία της κλάσης, όπως το αρχείο που θα χρησιμοποιηθεί για την εγγραφή και ανάγνωση των δεδομένων, το buffer που θα χρησιμοποιηθεί για την εγγραφή των δεδομένων, καθώς και τα offsets της τρέχουσας και τελευταίας εγγραφής.
- Ακολουθούν οι getters, setters των DiskAccesses, File f, lastWriteOffset, writeOffset.

- **Public void writeDataToFile():** εγγράφει έναν πίνακα αντικειμένων τύπου DataClass στο αρχείο.
- **Public byte[] read():** Διαβάζει δεδομένα από το αρχείο και τα επιστρέφει ως πίνακα αντικειμένων τύπου "DataClass". Η παράμετρος "dataSize" καθορίζει τον αριθμό των δεδομένων που θα διαβαστούν από το αρχείο.
- **retrieveData(int index):** Ανακτά δεδομένα από το αρχείο βάσει του δείκτη που δίνεται ως παράμετρος "index". Επιστρέφει ένα αντικείμενο τύπου "DataClass" που αντιπροσωπεύει τα δεδομένα που ανακτήθηκαν από το αρχείο, βασιζόμενο στον δείκτη "index".

DataPage:

- **DataPage():** Ο constructor της κλάσης που δημιουργεί ένα αντικείμενο τύπου "DataPage".
- **getBytes():** Επιστρέφει τα δεδομένα της σελίδας δεδομένων ως έναν πίνακα από bytes.

Μέθοδος B:

Η μέθοδος B αποτελείται από τις κλάσεις IndexFileClass, PairClass.

IndexFileClass:

- **IndexFileClass(File indexFile):** Αυτή η μέθοδος είναι ο constructor της κλάσης και δημιουργεί ένα αντικείμενο της κλάσης IndexFileClass.
- Ακολουθούν οι αντίστοιχοι getters, setters.
- **writeIndex(byte[] data):** εγγράφει έναν πίνακα bytes (data) στο αρχείο δεικτοδότησης (indexFile). Η διαδικασία γίνεται σε τμήματα μεγέθους DataPage.DataPageSize, όπου κάθε τμήμα αναπαριστά μια σελίδα δεδομένων.
- **readIndex(int dpOffset):** αναγνωρίζει ένα τμήμα του αρχείου δεικτοδότησης (indexFile) που αντιστοιχεί σε μια σελίδα δεδομένων βάσει του δείκτη dpOffset.

PairClass:

- **PairClass(int key, int pagePosition):** Αυτή η μέθοδος είναι ο constructor της κλάσης και δημιουργεί ένα αντικείμενο της κλάσης PairClass.

- Ακολουθούν οι αντίστοιχοι getters, setters.
- **createPair(DataClass[] dc):** Αυτή η μέθοδος χρησιμοποιείται για τη δημιουργία ενός πίνακα PairClass από έναν πίνακα DataClass.
- **savePairs(PairClass[] pc, IndexFileClass iFile):** Αυτή η μέθοδος χρησιμοποιείται για την αποθήκευση του πίνακα PairClass σε ένα αρχείο δεικτοδότησης (IndexFileClass).
- **pairToByteArray(PairClass[] dataArray):** Αυτή η μέθοδος χρησιμοποιείται για τη μετατροπή ενός πίνακα PairClass σε έναν πίνακα byte.
- **byteToPairArray(byte[] bytes):** χρησιμοποιείται για τον μετατροπή ενός πίνακα bytes σε έναν πίνακα αντικειμένων PairClass[].
- **searchKey(int key, IndexFileClass iFile):** χρησιμοποιείται για την αναζήτηση ενός κλειδιού (key) σε έναν πίνακα αντικειμένων της κλάσης PairClass που είναι αποθηκευμένος σε ένα αρχείο δείκτη (iFile).
- **compareTo :** υλοποιεί το αρθρωτό διάταξης (natural ordering) της κλάσης PairClass, η οποία υποδηλώνει το πώς οι αντικείμενα της κλάσης πρέπει να συγκρίνονται μεταξύ τους για την ταξινόμησή τους σε μια σειρά.

Μέθοδος Γ:

Η μέθοδος Γ αποτελείται από την κλάση BinarySearch.

- **BinarySearch(int[] data):** Αυτή η μέθοδος είναι ο constructor της κλάσης.
- Ακολουθούν οι αντίστοιχοι getters, setters.
- **search(int key):** Αυτή η μέθοδος υλοποιεί την αναζήτηση με δυαδική αναζήτηση στο αρχείο δεικτοδότησης (index file) για ένα κλειδί (key). Ελέγχει αν το αντικείμενο της κλάσης IndexFileClass είναι κενό (null) και επιστρέφει null αν είναι. Στη συνέχεια, καλεί τη μέθοδο doBSearch(int key) για να πραγματοποιήσει την πραγματική αναζήτηση.
- **doSearch(int key, IndexFileClass iFile):** Αυτή η μέθοδος υλοποιεί την αναζήτηση με δυαδική αναζήτηση στο αρχείο δεικτοδότησης (index file) για ένα κλειδί.

Ακολουθούν οι συμπληρωμένοι πίνακες των αποτελεσμάτων του κώδικα:

	A	A	B	B	Γ	Γ
--	---	---	---	---	---	---

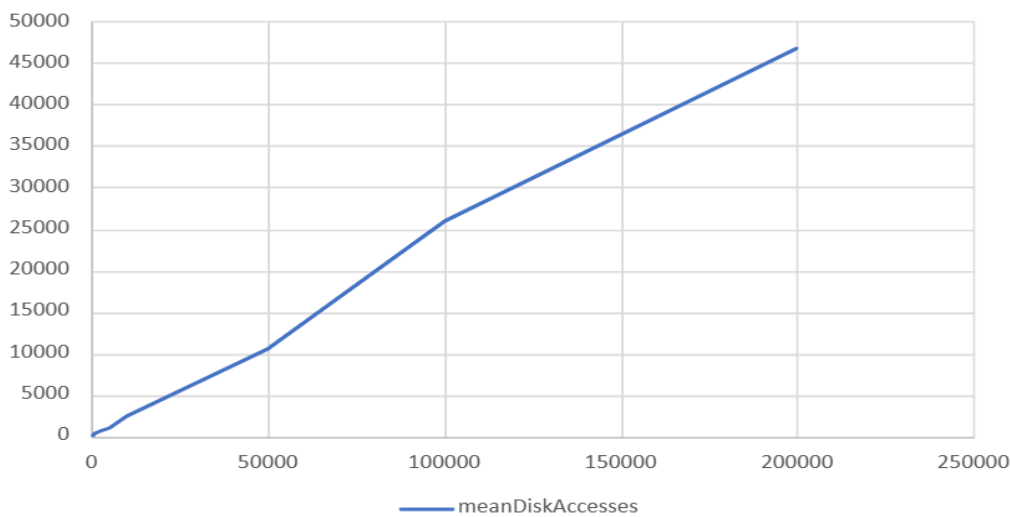
N	Disk Accesses (55bytes)	Disk Accesses (27bytes)	Disk Accesses (55bytes)	Disk Accesses (27bytes)	Disk Accesses (55bytes)	Disk Accesses (27bytes)
50	10.353	5.487	2.454	2.456	1.0	1.0
100	11.572	12.826	3.776	3.834	2.803	2.687
200	12.363	30.269	6.225	6.214	3.189	3.284
500	100.394	78.332	13.326	13.401	4.338	4.289
800	229.292	136.806	20.666	20.881	5.061	4.976
1000	356.462	181.642	24.704	24.901	5.132	5.09
2000	445.673	351.346	47.488	47.581	5.984	6.07
5000	970.053	788.906	119.559	117.158	7.381	7.443
10000	2308.281	1645.064	230.857	233.294	8.387	8.464
50000	10481.877	6201.905	1174.20 9	1187.591	10.751	10.61
100000	25867.049	13678.328	2300.37 7	2356.721	11.698	11.726
200000	46858.914	28745.637	4753.51	4734.678	12.702	12.694

	A	A	B	B	Γ	Γ
N	Χρόνος ανά αναζήτηση (55bytes)	Χρόνος ανά αναζήτηση (27bytes)	Χρόνος ανά αναζήτηση (55bytes)	Χρόνος ανά αναζήτηση (27bytes)	Χρόνος ανά αναζήτηση (55bytes)	Χρόνος ανά αναζήτηση (27bytes)
50	26642.1	9984.5	6060.7	4280.4	5153.5	8188.5
100	23759.0	22513.5	6751.8	6447.1	7925.0	9511.4
200	21235.6	55436.5	11240.7	10572.3	8426.8	10740.3
500	179196.7	167184.7	22605.1	29904.8	12230.6	12626.1
800	422329.88	241210.8	43149.5	34791.2	11129.0	13285.4
1000	570405.2	317799.72	46877.3	41052.0	11272.6	12626.1
2000	766540.25	637984.1	84530.9	78961.3	12592.1	13871.6
5000	15922524. 5	1431223.8	196124.8 9	195328.1	14911.6	16052.6

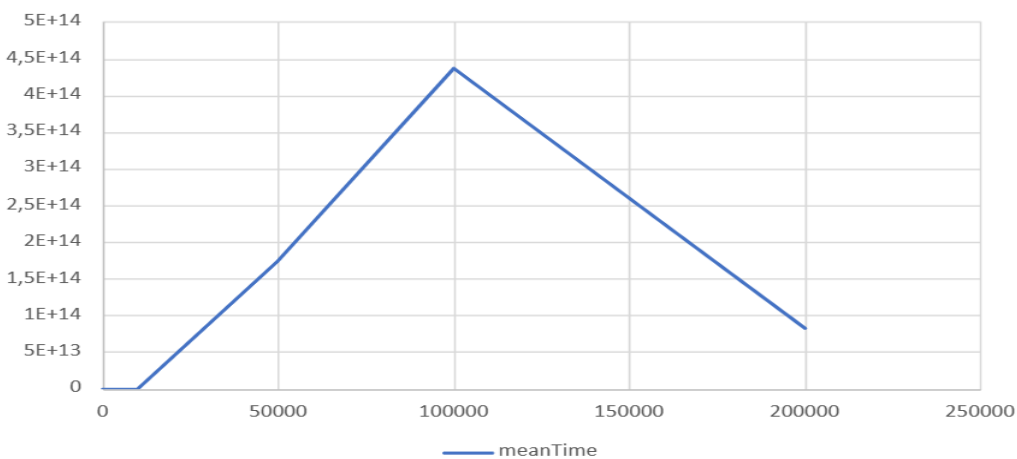
10000	3758061.2	3005830.0	382405.0	391157.4	16959.8	20390.9
50000	1.7324118 E7	1.1941379 E7	2031653. 4	2003559. 0	24538.1	21062.4
100000	4.3675584 E7	2.5314598 E7	3962309. 0	4173288. 5	22338.6	26557.4
200000	8.229597E 7	5.3087628 E7	4753.51	1.716944 E7	25414.9	27941.7

Ακολουθούν τα διαγράμματα που προκύπτουν από τα αποτελέσματα του κώδικα. Ο άξονας X αντιπροσωπεύει το πλήθος των στιγμιοτύπων και ο άξονας Y στον μέσο αριθμό προσβάσεων στον δίσκο και στον μέσο χρόνο εκτέλεσης αντίστοιχα.

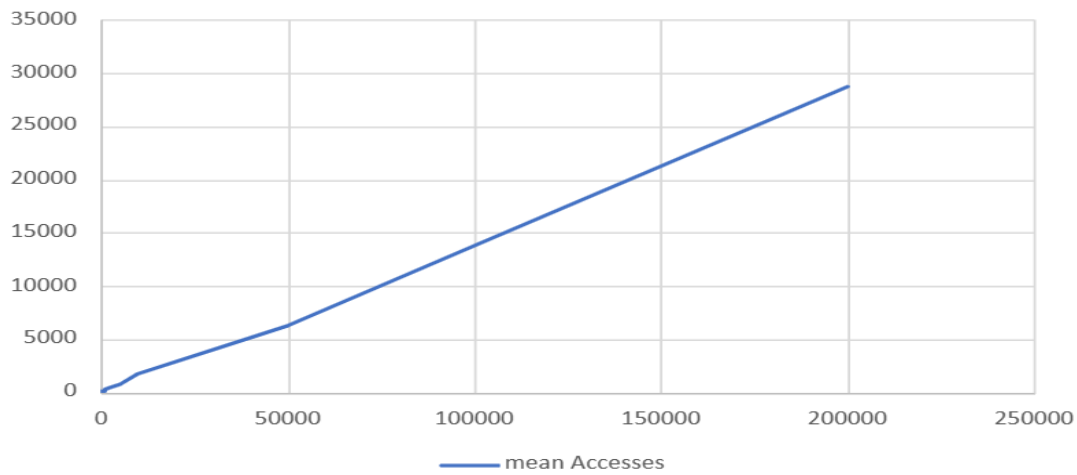
Method A: Disk Accesses(55 bytes)



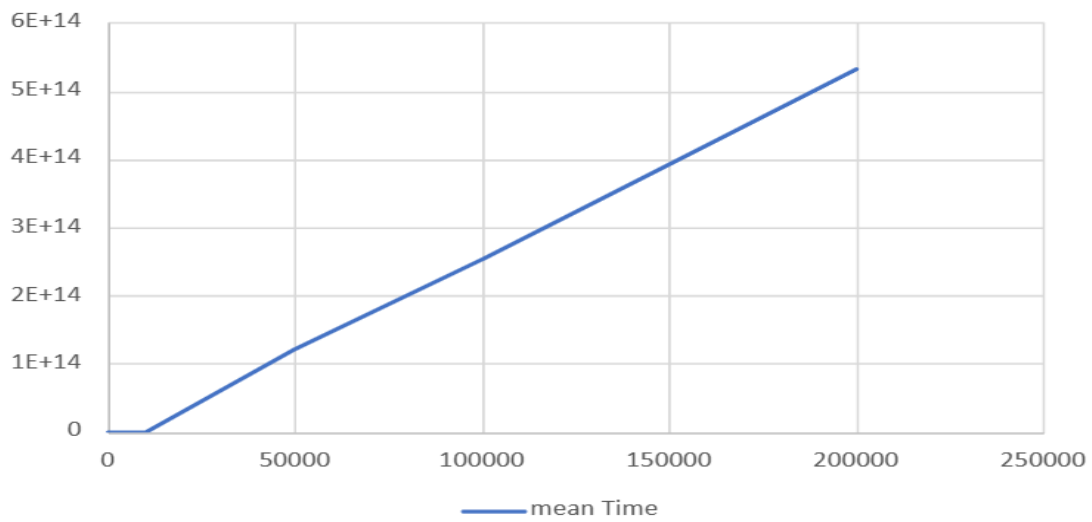
Method A: mean Time(55 bytes)



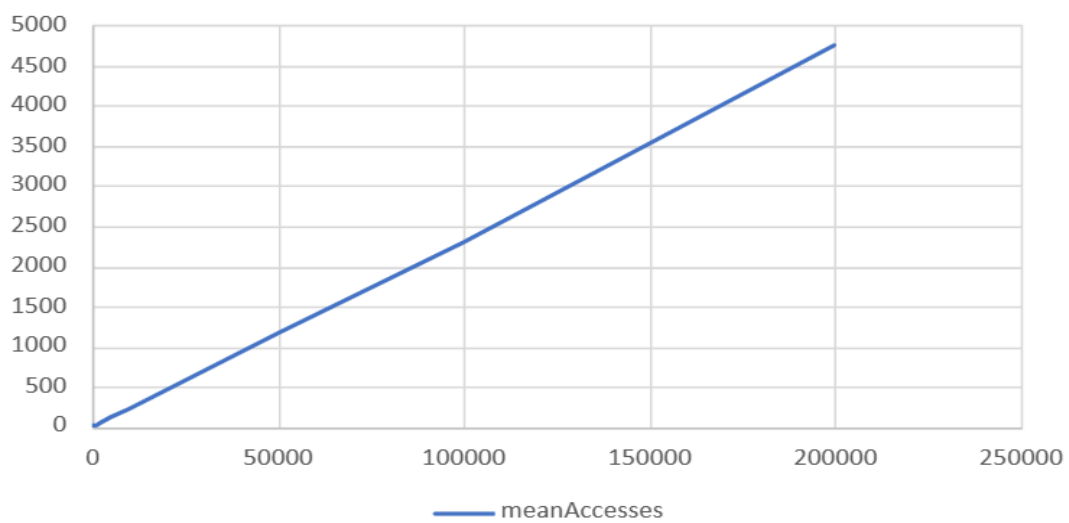
Method A: Disk Accesses(27 bytes)



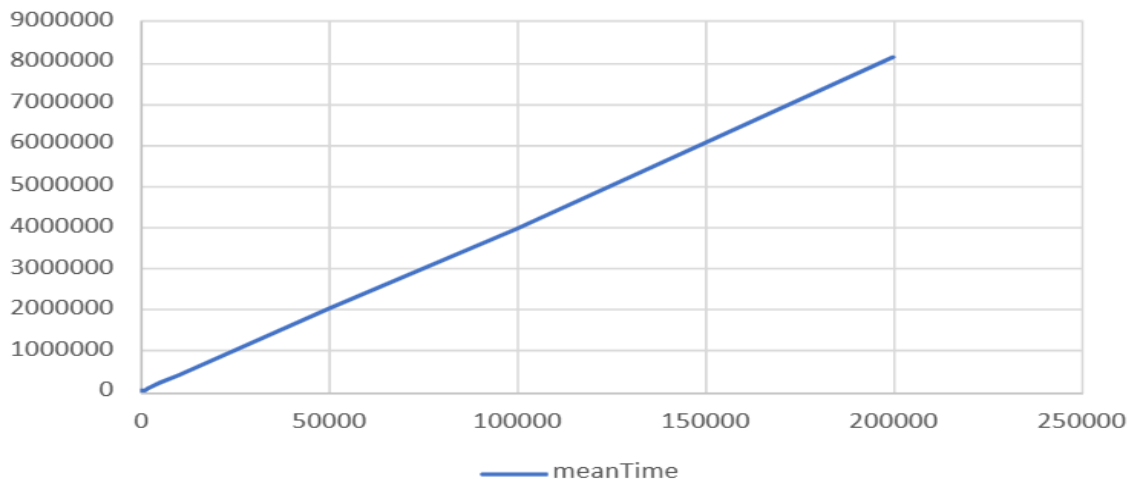
Method A: mean Time(27 bytes)



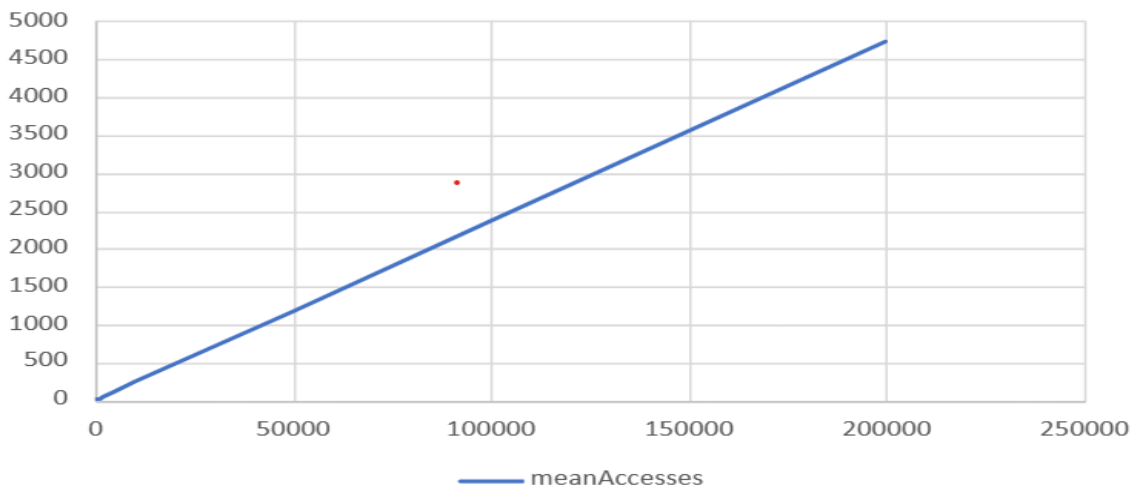
Method B: Disk Accesses(55 bytes)



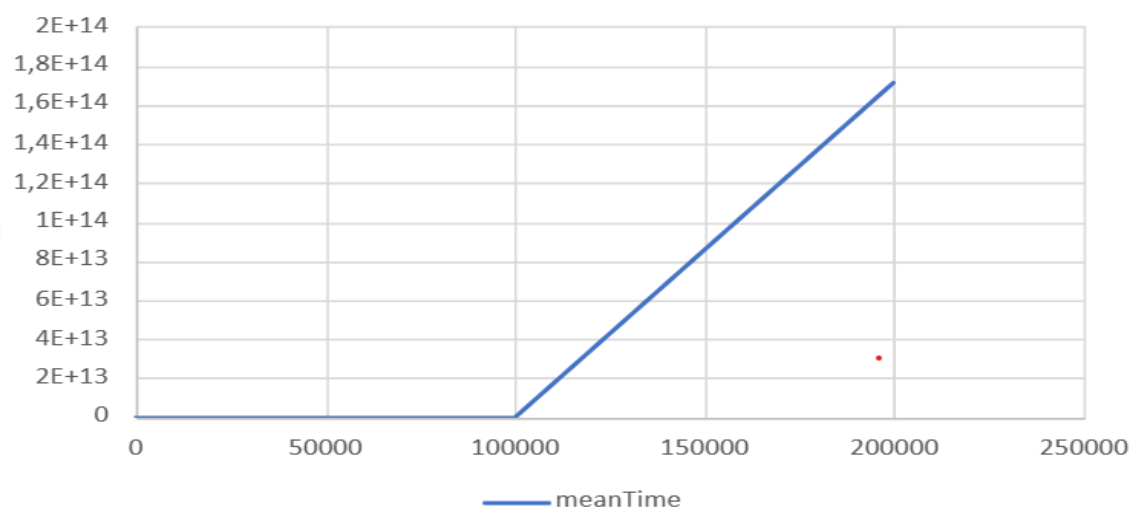
Method B: mean Time(55 bytes)



Method B: Disk Accesses(27 bytes)

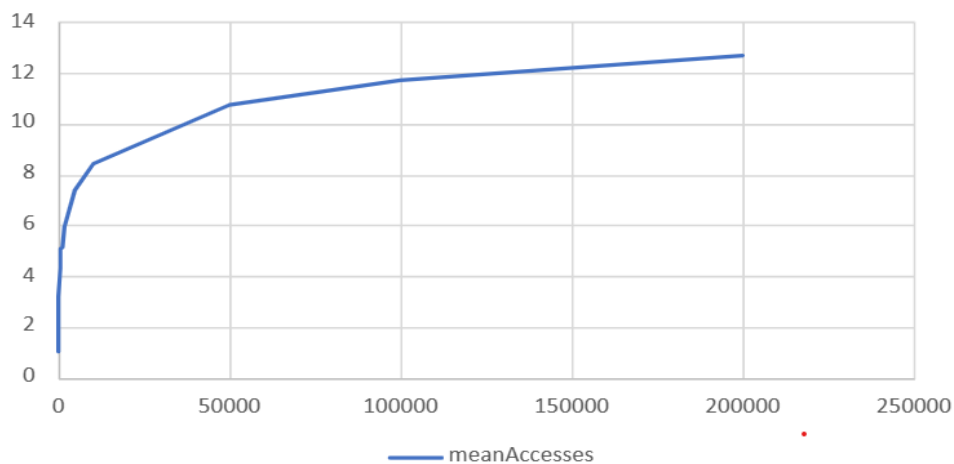


Method B: mean Time(27 bytes)

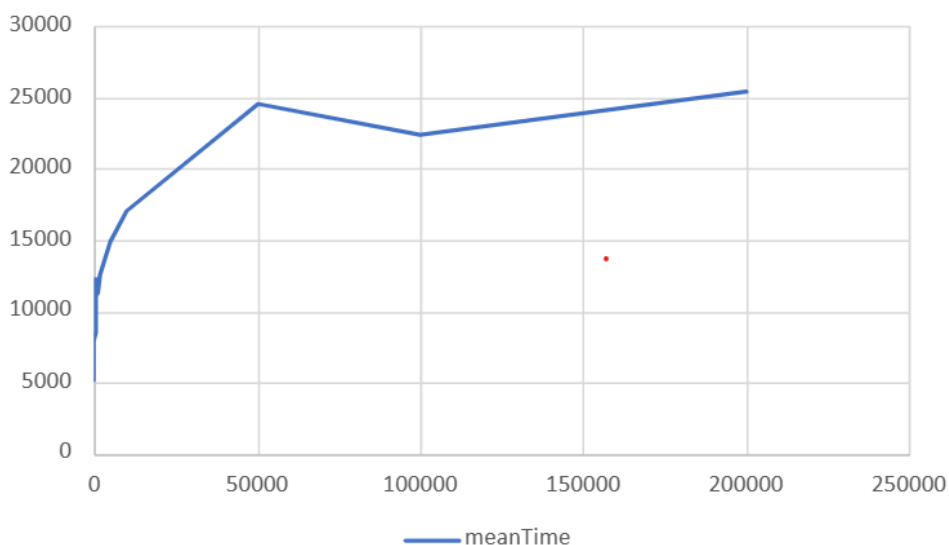


Σχόλια: Στις μεθόδους A και B οι καμπύλες των διαγραμμάτων είναι γραμμικές. Είναι, δηλαδή, ευθείες, οι οποίες συγκεκριμένα είναι ανοδικές. Αποτυπώνεται, έτσι, η σχέση μεταξύ των μεταβλητών, η οποία δηλώνει πως η αύξηση ή μείωση μίας μεταβλητής, συνήθως, συνδέεται με μια ανάλογη αύξηση ή μείωση της άλλης ανάλογα με τον προσανατολισμό της γραφικής παράστασης. Η μεγάλη κλίση της ευθείας υποδηλώνει μεγάλο ρυθμό αλλαγής των τιμών. Παρατηρείται, όμως, ότι στο διάγραμμα του χρόνου της μεθόδου A(55 bytes) η καμπύλη έχει ανοδική πορεία μέχρι τα 100.000 στιγμιότυπα και μετά είναι καθοδική. Αυτό υποδηλώνει αλλαγή στην κατεύθυνση της σχέσης μεταξύ των προσβάσεων στον δίσκο και του χρόνου εκτέλεσης. Στο διάγραμμα χρόνου της μεθόδου B (27 bytes), επίσης, παρατηρείται μια αρχική στασιμότητα της καμπύλης μέχρι τα 100.000 στιγμιότυπα και στη συνέχεια αυξάνεται.

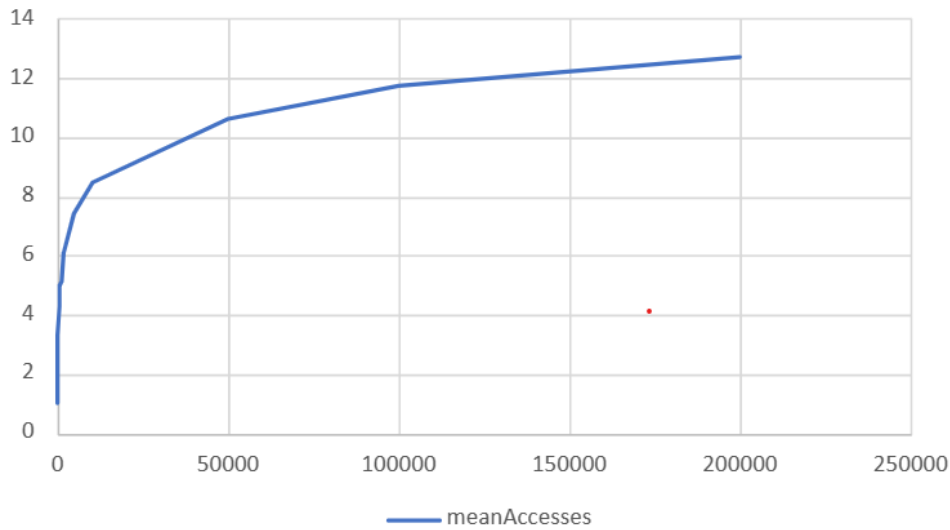
Method C: Disk Accesses(55 bytes)



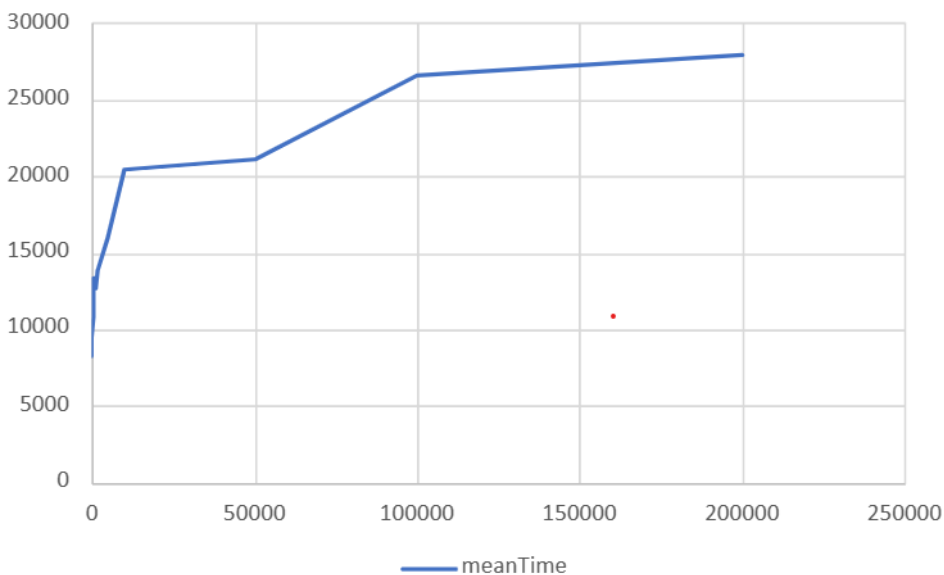
Method C: mean Time(55 bytes)



Method C: Disk Accesses(27 bytes)



Method C: mean Time(27 bytes)



Σχόλια: Στην μέθοδο C οι καμπύλες των διαγραμμάτων έχουν λογαριθμική μορφή. Η λογαριθμική μορφή του διαγράμματος μπορεί να υποδηλώνει μια αύξηση της τιμής της μεταβλητής που αναλύεται. Αυτό σημαίνει ότι, η αύξηση της τιμής είναι πιο γρήγορη καθώς προχωράμε προς το δεξί μέρος του διαγράμματος. Η γραφική παράσταση αλλάζει κλίση σε ορισμένα σημεία των καμπυλών, γεγονός που υποδηλώνει την αλλαγή του ρυθμού αύξησης ή μείωσης της μεταβλητής, δηλαδή των αριθμό προσβάσεων και τον χρόνο εκτέλεσης σε κάθε περίπτωση αντίστοιχα. Εφόσον, η μορφή είναι λογαριθμική δεν συνεπάγεται ότι η αλλαγή μίας μεταβλητής θα προκαλεί ανάλογη αλλαγή στην τιμή της άλλης μεταβλητής.

Αναφορά εξωτερικών πηγών:

Έγινε χρήση των παραδειγμάτων από τα φροντιστήρια του μαθήματος και από την ενότητα “Χρήσιμα” της εκφώνησης.

<https://www.geeksforgeeks.org/generate-random-string-of-given-size-in-java/>

[Binary Search in Java - GeeksforGeeks](#)

[Java String compareTo\(\) Method \(w3schools.com\)](#)

[Using Pairs in Java | Baeldung](#)

[FileInputStream \(Java Platform SE 7 \) \(oracle.com\)](#)

[FileOutputStream \(Java Platform SE 7 \) \(oracle.com\)](#)