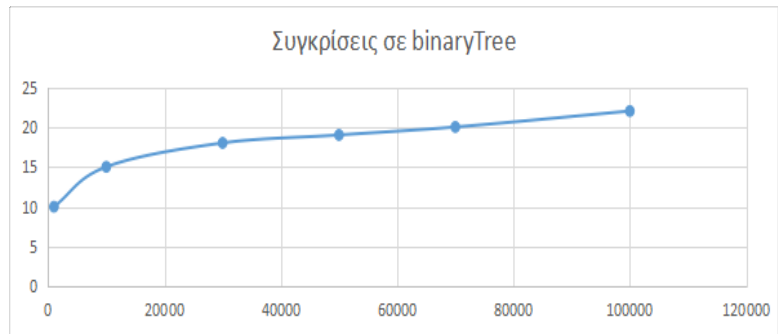


Έκθεση 3^{ης} Άσκησης

Δομές Δεδομένων και Αρχείων

Για το Διαδικό Δένδρο Έρευνας(BST):

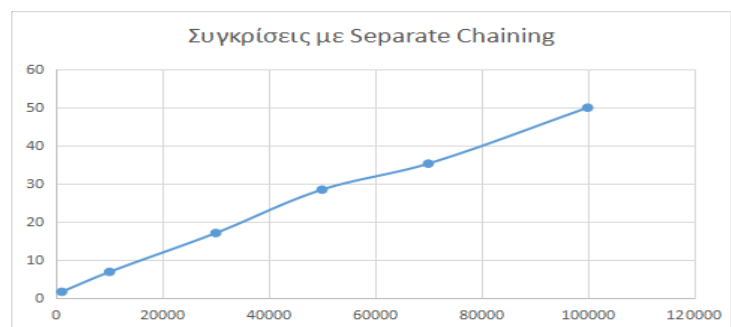
times	Comparisons average
1000	10
10000	15
30000	18
50000	19
70000	20
100000	22



Ο μέσος όρος των συγκρίσεων αυξάνεται με την αύξηση των δεδομένων με λογαριθμικό βαθμό. Από το αποτέλεσμα παρατηρούμε ότι η καμπύλη που προκύπτει μοιάζει με την καμπύλη της $\log N$ (base 2) με N τα δεδομένα που βάζουμε στο δέντρο, όπως και αναμένουμε. Πολυπλοκότητα σύμφωνα με την θεωρία $O(\log N \text{ base } 2)$.

Για το Separate Chaining:

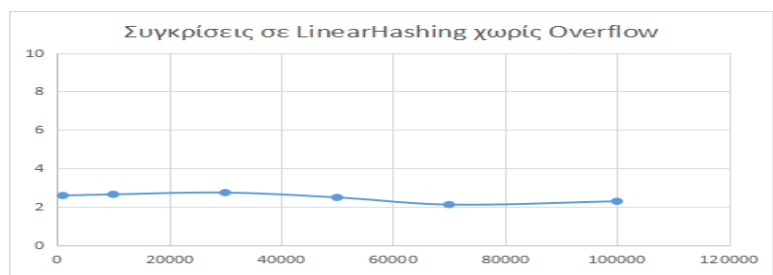
times	Comparisons average
1000	1,7
10000	6,9
30000	17,06
50000	28,4
70000	35,2
100000	49,83



Ο μέσος όρος των συγκρίσεων με αυτόν τον τρόπο αναζήτησης βλέπουμε ότι ανεβαίνει γραμμικά αυτή την φορά. Για πολύ μικρό ποσό δεδομένων έχουμε σχεδόν μία αναζήτηση αλλά όσο τα δεδομένα πληθαίνουν, οι αναζητήσεις αυξάνονται ραγδαία, κάτι το οποίο και περιμέναμε σύμφωνα με την θεωρία. Πολυπλοκότητα σύμφωνα την θεωρία από $O(1)$ - $O(N)$.

Για το Linear Hashing χωρίς Overflow:

times	Comparison s average
1000	2,6
10000	2,66
30000	2,75
50000	2,5
70000	2,13

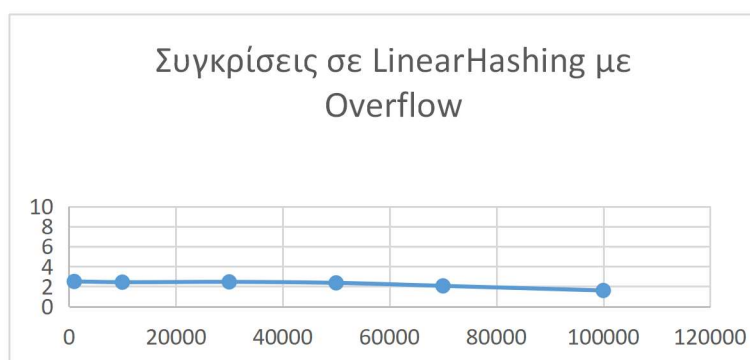


Ο μέσος όρος των συγκρίσεων με αυτόν το τρόπο αναζήτησης βλέπουμε ότι κυμαίνεται γύρω από το 2,6 (με αποκλίσεις που θεωρούνται μηδαμινές) για ανεξάρτητο αριθμό δεδομένων. Επομένως, όπως

αναμένουμε, η γραφική που απεικονίζει τις συγκεκριμένες τιμές του πίνακα είναι σχεδόν μια ευθεία γραμμή με πολύ λίγες αναζητήσεις, κάτι το οποίο όπως ξέρουμε χαρακτηρίζει το Linear Hashing. Πολυπλοκότητα σύμφωνα με την θεωρία $O(1)$.

Για το Linear Hashing με Overflow:

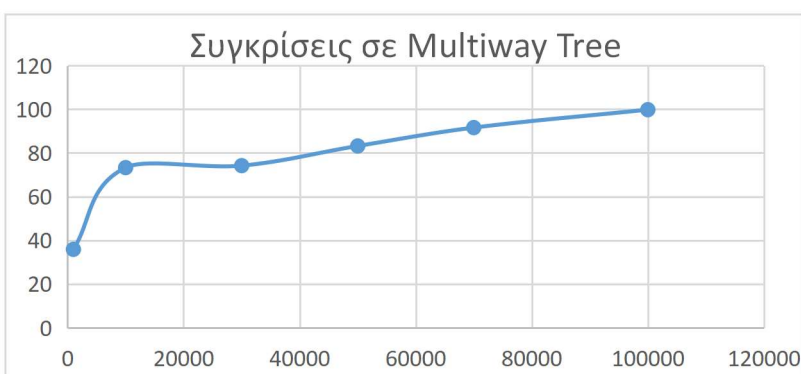
times	Comparisons average
1000	2,5
10000	2,43
30000	2,46
50000	2,36
70000	2,06



Σε αυτήν την περίπτωση Linear Hashing βλέπουμε πάλι ότι η τιμή κυμαίνεται κοντά στο 2,5(και πάλι με ελάχιστες αποκλίσεις για ανεξάρτητο αριθμό δεδομένων). Βλέπουμε παρόλα αυτά, ότι για μεγάλο αριθμό δεδομένων οι συγκρίσεις σε σχέση με το Linear Hashing χωρίς Overflow είναι ελάχιστα λιγότερες, όπως είναι και αναμενόμενο μιας και που με το Overflow η διάσπαση γίνεται περισσότερες φορές. Η γραφική που απεικονίζει τον συγκεκριμένο τρόπο αναζήτησης τείνει να πλησιάσει την ευθεία γραμμή. Πολυπλοκότητα σύμφωνα με την θεωρία $O(1)$.

Για το Multiway Tree:

times	Mtree
1000	35,9
10000	73,3
30000	74,2
50000	83,2
70000	91,6



Τέλος, για τον τρόπο αναζήτησης Multi way Tree βλέπουμε ότι οι συγκρίσεις που χρειάζεται να γίνουν ακόμα και για μικρό αριθμό δεδομένων είναι αρκετές. Ο λόγος είναι ότι από την δομή του έχει πάρα πολλές τιμές σε κάθε κόμβο που πρέπει να διασχίσει. Παρόλα αυτά βλέπουμε και πάλι μέσω της γραφικής ότι προκύπτει καμπύλη που τείνει στην λογαριθμική. Πολυπλοκότητα σύμφωνα με την θεωρία $O(\log N \text{ base } 100)$.

Συγκριση Δομών Βάσει των Αποτελεσμάτων:

Με βάση λοιπόν την συνοπτική (λόγω του δεδομένου πλήθους σελίδων που μας δόθηκε) παρουσίαση των δομών δεδομένων, καταλήγουμε, όπως αναμέναμε, ότι για πολύ μεγάλο αριθμό δεδομένων προτιμούμε το Linear Hashing με το Overflow καθώς τα δεδομένα σύμφωνα με την θεωρία χρειάζονται ελάχιστες αναζητήσεις όπως και αποδεικνύεται από τις τιμές των μέσων όρων. Το Linear Hashing χωρίς Overflow έρχεται δεύτερο, με ελάχιστες διαφορές, οπότε αυτομάτως καθίσταται και αυτό σαν μια πολύ καλή επιλογή, ειδικά για μεγάλο αριθμό δεδομένων. Επιπλέον, για πολύ μικρό αριθμό δεδομένων παρατηρούμε ότι και το separate Chaining είναι η καλύτερη επιλογή μιας και σύμφωνα με τις μετρήσεις που κατέχουμε έχει για πολύ λίγα δεδομένα τον μικρότερο αριθμό συγκρίσεων κατα μέσο όρο. Τέλος τα δέντρα λόγω της δομής τους, μας δίνουν γενικά πολυπλοκότητα λογαρίθμου οπότε γενικά τα αποφεύγουμε.

Πηγές:

Για το δυαδικό δέντρο:

Απόλυτη Java, εκδόσεις Ιων επιμέλεια Δημήτρης Ιακωβίδης

Για το seperate Chaining

http://www.java2s.com/Code/Java/Collections-Data-Structure/Hashtablewithseparatechaining.htm?fbclid=IwAR0h0iCVm7yhsj6Fi75V9mlp6_pD0fPVpDyIXadLIQ5fkUY1PH5fXK73ams

Για το Linear Hashing

Courses.ece.tuc.gr

Multiway Tree

<https://www.geeksforgeeks.org/b-tree-set-1-insert-2/>