

Πανεπιστήμιο Πατρών
Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Δομές Δεδομένων
Εργαστηριακή Άσκηση
Ακαδημαϊκό Έτος 2017 – 2018

Μονοπάτης Δημήτριος

ΑΜ: 4776

monopatis@ceid.upatras.gr

Η άσκηση υλοποιήθηκε σε γλώσσα C σε λειτουργικό με πυρήνα GNU/Linux. Η μεταγλώττιση των πηγαίων αρχείων μπορεί να γίνει με εκτέλεση της εντολής “make”.

Μέρος Α (partA.c)

Αρχικά γίνεται αποθήκευση των αριθμών από το αρχείο intergers.txt σε μια δομή Array που διπλασιάζει τον χώρο που καταλαμβάνει στην μνήμη προοδευτικά. Εμφανίζονται τα 10 πρώτα στοιχεία του και τα 10 τελευταία και εφαρμόζεται ο αλγόριθμος Merge Sort. Τέλος γίνεται αποδέσμευση της μνήμης.

Μέρος Β (partB.c)

Επέκταση του προηγούμενου προγράμματος. Γίνεται επιλογή του αλγορίθμου αναζήτησης και του στοιχείου αναζήτησης από το πληκτρολόγιο ή κατά την κλήση του προγράμματος με δεύτερο όρισμα το στοιχείο προς αναζήτηση και τρίτο τον τρόπο (1-3) (1=Γραμμική Αναζήτηση 2=Binary search 3=Interpolation search)

πχ “./partB 12334 2”

αναζήτηση του 12334 με εφαρμογή Binary search

Στο Interpolation search είναι απαραίτητο να γίνει casting με double στην επιλογή του επόμενου στοιχείου

```
int next = l + ((double)r - l) * (searching_el - a->array[l]) / (a->array[r] - a->array[l]);
```

Μέρος Γ (partC.c)

Έγινε υλοποίηση του Red-Black Tree.

Σημαντικό σημείο βρίσκεται στις γραμμές 369-370 που επαληθεύεται πως η ρίζα είναι μαύρη και δεν έχει πατέρα, κάτι που χάλασε σε περίπτωση περιστροφής της ρίζας.

Μέρος Δ (partD.c)

Ο χειρότερος χρόνος στη γραμμική αναζήτηση είναι της τάξης $O(n)$ και εμφανίζεται όταν αναζητούμε το τελευταίο στοιχείο ή κάποιο αριθμό μεγαλύτερο από αυτό. Και ο μέσος χρόνος είναι $O(n)$.

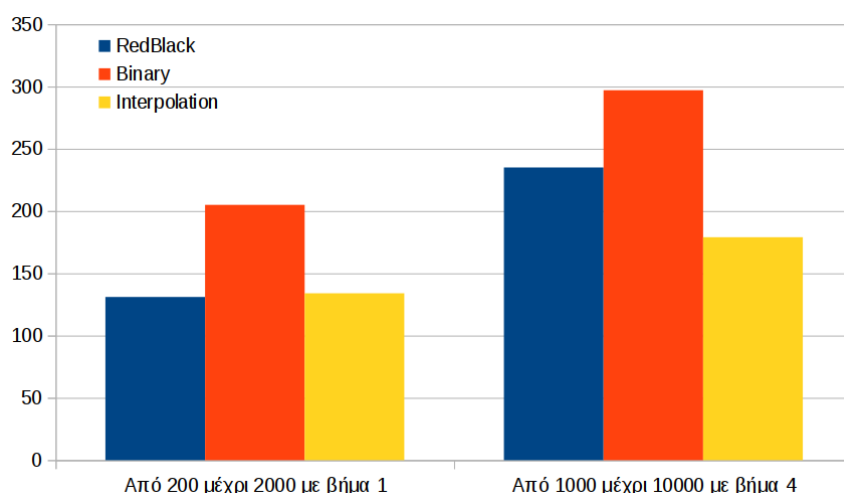
Η Binary search (δυϊκή αναζήτηση) στην χειρότερη περίπτωση χρειάζεται χρόνο $O(n \log n)$ και μπορεί να εμφανιστεί σε πολλές περιπτώσεις πχ όταν αναζητούμε το τελευταίο στοιχείο ή κάποιο αριθμό μεγαλύτερο από αυτό ή το πρώτο. Ενώ ο μέσος χρόνος είναι $O(\log n)$

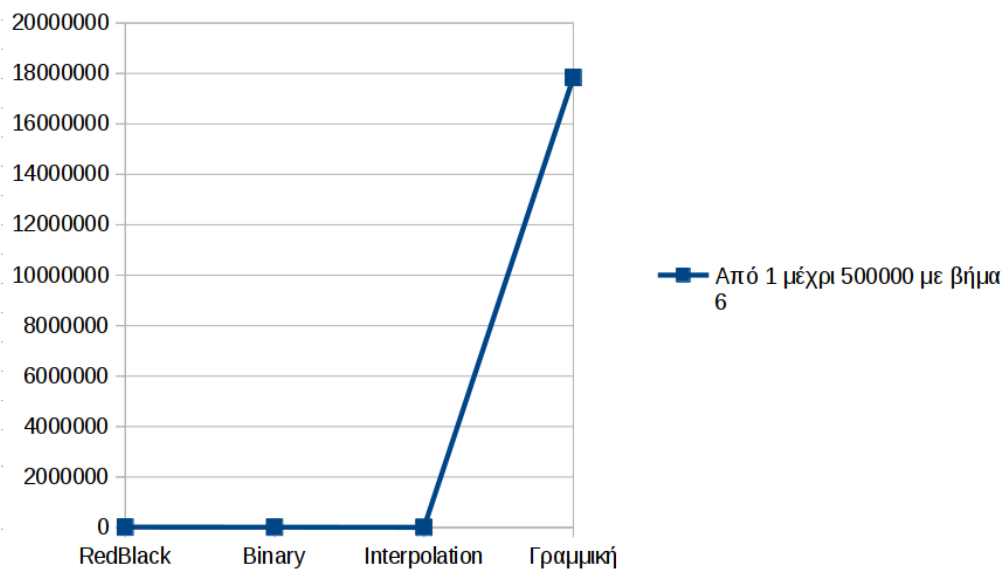
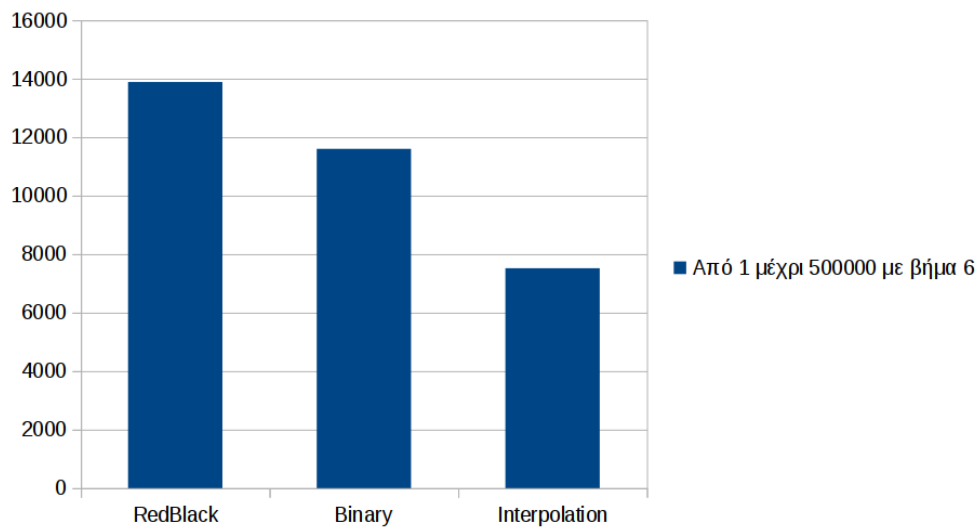
Η Interpolation search στην χειρότερη περίπτωση χρειάζεται χρόνο $O(n)$ ενώ ο μέσος χρόνος είναι $O(\log \log n)$

Η αναζήτηση σε ένα Red-Black Δέντρο είναι της τάξης $O(\log n)$ και για μέση και για χειρότερη περίπτωση.

Ακολουθούν οι χρόνοι για κάποια πειράματα σε microseconds:

Αναζήτηση	RedBlack	Γραμμική	Binary	Interpolation
του 10	1	0	1	0
του 500000	0	238	0	0
Από 1 μέχρι 500000 με βήμα 6	13887	17837436	11603	7520
Από 200 μέχρι 2000 με βήμα 1	131	342071	205	134
Από 1000 μέχρι 10000 με βήμα 4	235	442940	297	179





Το interpolation search είναι η πιο γρήγορη εκτός από ένα πείραμα που ήταν η αναζήτηση στο Red-Black. Η γραμμική η πιο αργή ειδικά όταν αναζητούνται στοιχεία που βρίσκονται προς το τέλος των στοιχείων. Η binary για πολλές αναζητήσεις είναι πιο γρήγορη από την αναζήτηση στο Red-Black, ενώ σε κάποια πειράματα πιο αργή από αυτήν.

Οπότε οι πραγματικοί χρόνοι συνάδουν με τους θεωρητικούς.

Μέρος Ε (partE.c)

Έγινε υλοποίηση του Digital Tree (TRIE)