

Περίδης Γιάννης 2018030069

HMMY , Δομές Δεδομένων και Αρχείων

2^η Εγαστηριακή άσκηση

Αναφορά Αποτελεσμάτων:

Πίνακας μετρήσεων: για $N=10^6$

Μέθοδος	Μέσος αριθμός συγκρίσεων / εισαγωγή	Συνολικός χρόνος για N εισαγωγές (ms)	Μέσος αριθμός συγκρίσεων / τυχαία αναζήτηση	Συνολικός χρόνος για N αναζητήσεις (ms)	Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=100)	Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=1.000)
ΔΔΕ με δυναμική παραχώρηση μνήμης	92,29	11162,66	82,54	0,8	127,04	128,28
ΔΔΕ με array	119,68	10968,37	82,54	4,28	124,82	127,73
Ταξινομημένο Πεδίο			102,9	0,31	-	-

(Στα δύο τελευταία κελία δεν μπόρεσα να βρώ τιμές όπως θα αναφαίρω παρακάτω αναλυτικότερα)

Παρατηρώ πως κατά μέσο όρο ο αριθμός συγκρίσεων κατά την εισαγωγή είναι αρκετά μεγαλύτερος στον τρόπο υλοποίησης με array, σε σχέση με την δυναμική παραχώρηση της μνήμης. Αντιθέτως, ο μέσος αριθμός συγκρίσεων σε μια τυχαία αναζήτηση είναι ακριβώς ο ίδιος, εφόσον και χρησιμοποιήθηκε στα static και στο dynamic για την υλοποίηση τους πολύ παρόμοιος κώδικας. Ακόμη, φαίνεται πως ο συνολικός χρόνος των N εισαγωγών, είναι αρκετά παρόμοιος στις δύο περιπτώσεις στο κομμάτι αυτών των συγκρίσεων με διαφορά κάποια ms παραπάνω στην δυναμική. Ενώ αντιθέτως ο χρόνος αναζήτησης με array είναι εξαιρετικά μεγαλύτερος από ότι στην δυναμική (καθώς και δεν έχει σημασία ότι είναι κάποια ελάχιστα ms, ποσοτικά είναι περίπου ο πενταπλάσιος), πράγμα που μπορεί να ωφείλεται σε κάποια δυσλειτουργία του προγράμματος μου. Τέλος για τις αναζητήσεις με μεγάλο εύρος, παρατηρώ μικρές διαφορές (ελάχιστα περισσότερες η δυναμική) οι οποίες τείνουν να εξαλείφονται όσο μεγαλώνει το εύρος αυτό.

Ανάλυση κώδικα:

Για το συγκεκριμένο project, μας ζητήθηκε να υλοποιήσουμε εμ τρίς διαφορετικούς τρόπους την δομή binary search tree(μια με δυναμική παραχώρηση μνήμης, μια με έναν πίνακα δηλαδή array δύο διαστάσεων και μια με ένα ταξινομημένο array), καθώς και να πραγματοποιήσουμε συγκεκριμένες μετρήσεις για κάθε μια από αυτές τις υλοποιήσεις.

Αρχικά, να αναφέρω για την διευκόλυνσή σας πως όπως έχω αναφέρει και στα σχόλια του κώδικά μου , ότι για να τρέξεις το πρόγραμμα πρέπει να πάς στο: Run as->Configurations->Arguments και να εισάγεις εκεί το path του αρχείου ,δηλαδή το location του.Ακόμη, αν θελήσετε να ξανατρέξετε το πρόγραμμα με κάποια άλλη τιμή εκτός του $N=10^6$, θα πρέπει οι ίδιοι να αλλάξετε την constant παράμετρο N που έχω ορίσει στην main για την σωστή λειτουργία του προγράμματος.

Ξεκινώντας την υλοποίηση του κώδικα,δημιούργησα μια διεπαφή η οποία περιέργαφε όλες τις αναγκαίες μεθόδους ενός binary search tree.Την διεπαφή αυτή την υλοποιούσαν δύο κλάσεις η BinarySearchTreeStatic (για την υλοποίηση με τον πίνακα) και η BinarySearchTreeDynamic (για την υλοποίηση με δυναμικής παραχώρησης μνήμης) , οι οποίες αποτέλεσαν και τις δύο πρώτες υλοποιήσεις μου.Οι δύο αυτές κλάσεις ήταν παρόμοιες σε πολλαπλά σημεία του κώδικα καθώς και χρησιμοποιηθηκαν οι ίδιες μέθοδοι για το binary search tree , με διαφοροποιήσεις φυσικά στο σώμα, για να ικανοποιηθούν τα ζητούμενα.Η τρίτη κλάση που έφτιαξα είναι το ArraySorted(για την υλοποίηση με το ταξινομημένο array).Αντίθετα, η τρίτη υλοποίηση διαφέρει απί τις άλλες δύο καθώς και είναι και εξαρτόμενη και από την μία από αυτές.Αυτό διότι η κλάση αυτή , αρχικοποιείται μέσω της BinarySearchTreeDynamic , με την βοήθεια της συνάρτησης inorder η οποία δημιουργεί ένα array που περιέχει τα keys της BinarySearchTreeDynamic κατανεμημένα σε αύξουσα σειρά.

Όσον αναφορά τις συναρτήσεις των δύο binary search tree κλάσεων ,δεν είναι απαραίτητη η μακριγορία και η ανάλυση τους καθώς και αυτές που πραγματοποιούν τις κυριότερες λειτουργίες δηλαδή insert,delete,search,inorder και range τι σβρίκα έτοιμες στα links που θα αναφάιρω παρακάτω, με φυσικά ορισμένες τροποποιήσεις για να συμβαδίσουν με την λειτουργικότητα που ήθελα ή για λόγους ωραιοποίησης του κώδικα.

Εδώ πρέπει να αναφέρω πως δεν κατάφερα να υλοποιήσω το κομμάτι του κώδικα της συνάρτησης range στην κλάση ArraySorted και έτσι κατά συνέπεια δεν είχα κάποια αποτελέσματα στα τελευταία δύο κελία του αρχικού μου πίνακα.

Οι πηγές που χρησιμοποίησα κυρίως για τις δύο κλάσεις του binary search tree είναι ένας συνδιασμός των παρακάτω , φυσικά πάντοτε με τις απαραίτητες αλλαγές:

<https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/>

<https://www.geeksforgeeks.org/print-bst-keys-in-the-given-range>

<https://stackoverflow.com/questions/6135668/reading-integer-values-from-binary-file-using-java>

Τέλος συμβουλευτικά το stackOverflow για να αντιμετωπίσω ορισμένα error που αντιμετωπίζα.