



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

ΜΑΘΗΜΑ: ΠΛΗ 201 – ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ
ΑΡΧΕΙΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

Αναφορά 2^{ου} Εργαστηρίου :

Όνομα: Παπουτσάκης Νικόλαος 2019030206

Εισαγωγή :

Στο 2^ο εργαστήριο ήρθαμε σε επαφή με μία καινούργια δομή δεδομένων, τα δυαδικά δέντρα. Για την εύκολη κατανόηση του τι γίνεται και του τι θέλουμε να κάνουμε σε αυτήν την άσκηση, θα χωρίσουμε το συγκεκριμένο project σε 2 μέρη. Πρώτον, την υλοποίηση σημαντικών «εργαλείων» που θα βοηθήσουν στην επεξεργασία των μετρήσεων και δεύτερον, το Search Testing, με το οποίο θα πραγματοποιήσουμε κάποια search tests για να συλλέξουμε κάποια δεδομένα που θέλουμε.

1^η Φάση : Υλοποίηση Δομής Δεδομένων

Αρχικά δημιουργήσαμε τις κλάσεις **Simple Binary Search Tree** και **Threaded Binary Search Tree** οι οποίες περιέχουν μεθόδους insert, find, print και findRange, απαραίτητες για το υπόλοιπο κομμάτι της άσκησης.

Για το 3^ο μέρος του test που χρειαζόμασταν ένα ταξινομημένο πεδίο και μια μέθοδο που θα έκανε binary search σε αυτό, δημιουργήσαμε τις κλάσεις Binary Search και Sorted Array που μας ήταν χρήσιμες. Πιο συγκεκριμένα,

- Η **Binary Search** παίρνει το ταξινομημένο πεδίο και εφαρμόζει τις μεθόδους search και search Range.
- Η **Sorted Array** παίρνει σαν όρισμα στον Constructor τα κλειδιά που έχουμε εισάγει στο δυαδικό δέντρο και με τη μέθοδο sortKeys τα ταξινομεί κατά αύξουσα σειρά και τα επιστρέφει.

2^η Φάση : Search Testing

Αφού δημιουργήσαμε τις παραπάνω κλάσεις έπειτα κληθήκαμε να δημιουργήσουμε κάποια testers με τα οποία θα ξεκινούσαμε τις αναζητήσεις που μας αναφέρει η εκφώνηση της άσκησης. Έτσι δημιουργήσαμε τις κλάσεις Simple Binary Tree Tester, Threaded Binary Tree Tester και Sorted Array Tester.

Κάθε μία από τις παραπάνω κλάσεις περιέχει μία μέθοδο do Test() η οποία εκτελεί τα τεστ, δηλαδή αρχικά μετράει το μέσο αριθμό συγκρίσεων ανά εισαγωγή (εκτός από την τελευταία περίπτωση με το ταξινομημένο πεδίο που δεν νοείται εισαγωγή). Έπειτα εκτελεί 100 αναζητήσεις τυχαίων αριθμών κλειδιών από 1 έως 10^6 και μετράει το μέσο αριθμό συγκρίσεων ανά αναζήτηση και τέλος εκτελεί 100 τυχαίες αναζητήσεις εύρους τιμών και μετράει το μέσο αριθμό συγκρίσεων ανά αναζήτηση.

Απόδοση και Τεκμηρίωση Αποτελεσμάτων:

Μέθοδος	Μέσος αριθμός συγκρίσεων / εισαγωγή	Μέσος αριθμός συγκρίσεων / τυχαία αναζήτηση	Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=100)	Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=1000)
ΔΔΕ Α	89	54	95	448
Νηματοειδές ΔΔΕ Β	116	80	149	680
Ταξινομημένο πεδίο		66	139	405

Σχόλια:

- Στην εισαγωγή προφανώς θα έχουμε περισσότερες συγκρίσεις στην περίπτωση του Threaded δέντρου αφού χρειάζεται να φτιάξουμε τα threads(και επίσης ο αλγόριθμος στο threaded tree δεν έχει αναδρομή).
- Στην αναζήτηση παρατηρούμε πως το απλό δέντρο έχει πάρει την 1^η θέση με τις λιγότερες συγκρίσεις και αμέσως μετά στο sorted array με την εφαρμογή της δυαδικής αναζήτησης.(Παρόλο που γνωρίζουμε ότι το time complexity και των 2 είναι στην καλύτερη $O(\log n)$).
- Στις αναζητήσεις εύρους τιμών παρατηρούμε πως στο threaded tree ο μέσος αριθμός είναι αυξημένος. Αυτό συμβαίνει καθώς ο αλγόριθμος δεν έχει γραφτεί με αναδρομή.
- Στο binary search αφού βρούμε το K1 μετά είναι εύκολο να βρούμε το K2 αφού απλά διασχίζουμε ένα ταξινομημένο πεδίο.