

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

2η άσκηση

Ημερομηνία παράδοσης: 25 Απριλίου 2020

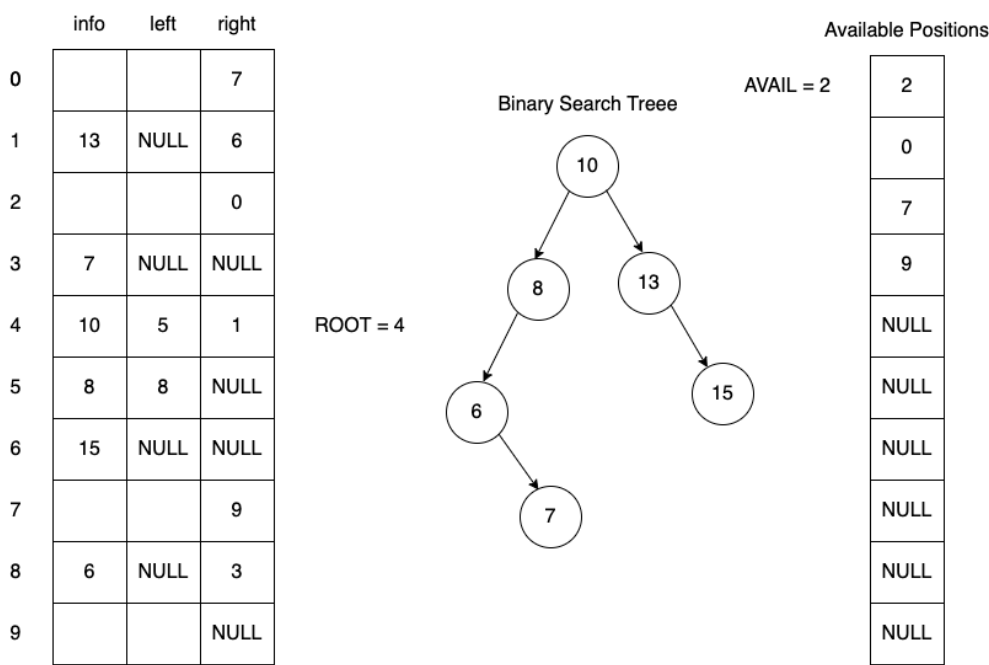
Η άσκηση είναι ατομική

Δυαδικό Δένδρο Έρευνας (4 μονάδες)

Κατασκευάστε την κλάση «Δυαδικό Δένδρο Έρευνας» (ΔΔΕ) με χρήση πεδίου αριθμών (array) μεγέθους N. Εκτός από τις βασικές πράξεις info(tree_pointer), left(tree_pointer), right(tree_pointer) που ισχύουν γενικά για ΔΔΕ, η κλάση υποστηρίζει τις πράξεις:

- Εισαγωγή τυχαίου κλειδιού
- Αναζήτηση τυχαίου κλειδιού
- Διάσχιση inorder
- Αναζήτηση εύρους τιμών¹.

Όπως συμβαίνει και στην περίπτωση συνδεδεμένης λίστας, η υλοποίηση μπορεί να γίνει είτε με δυναμική παραχώρηση μνήμης είτε με πεδίο (array). Για την υλοποίηση του ΔΔΕ χρησιμοποιείστε ένα πεδίο Nx3 ακεραίων αριθμών (N είναι ο αριθμός των στοιχείων). Κάθε θέση του πίνακα παριστάνει έναν κόμβο του δένδρου και σε κάθε μία από τις 3 στήλες αποθηκεύονται τα πεδία info, left, right του κόμβου².



¹ <https://www.geeksforgeeks.org/print-bst-keys-in-the-given-range/>

² Ένα πεδίο μεγέθους N στοιχείων μπορεί να παραστήσει ένα ΔΔΕ με N στοιχεία το πολύ.

Το παραπάνω σχήμα (αριστερά), δείχνει της υλοποίηση του ΔΔΕ (κέντρο). Η ρίζα του δένδρου βρίσκεται στην θέση 4 (ROOT= 4). Η υλοποίηση είναι αντίστοιχη με την υλοποίηση συνδεδεμένης λίστας με array³. Το πεδίο left έχει τιμή που δηλώνει την θέση του αριστερού υποδένδρου. Το πεδίο right έχει διπλό ρόλο: (α) δηλώνει την θέση του δεξιού υποδένδρου και (β) χρησιμοποιείται επίσης για να υλοποιήσει την στοίβα (stack) με τις διαθέσιμες θέσεις του array. Στο σχήμα (δεξιά), η θέση AVAIL=2 δηλώνει ότι θέση 2 είναι η επόμενη διαθέσιμη θέση που θα χρησιμοποιηθεί στην επόμενη εισαγωγή κόμβου. Αν γίνει διαγραφή κόμβου, τότε αυτή θα προστεθεί στην κορυφή της στοίβας (και η θέση AVAIL θα πάρει τιμή την θέση του κόμβου που διαγράφηκε). Επομένως, για να υλοποιήσετε τις μεθόδους εισαγωγής και διαγραφής στοιχείων από τον ΔΔΕ θα χρειαστείτε πιο πριν να έχετε υλοποιήσει της μεθόδους getnode(tree_pointer) και free_node(tree_pointer) στην στοίβα των ελεύθερων θέσεων του array⁴. Στο παραπάνω σχήμα (δεξιά) φαίνεται η μορφή της στοίβας που υλοποιεί η στήλη right του array . Προσέξτε ότι δεν είναι ανάγκη να φτιάξετε την στοίβα με ξεχωριστή δομή δεδομένων έξω από το array (η στοίβα στα δεξιά του σχήματος υπάρχει ήδη στο πεδίο right).

Απόδοση Δυαδικού Δένδρου Έρευνας (4 μονάδες)

Χρησιμοποιήστε ένα πεδίο με Nx3 με N θέσεις και κάντε εισαγωγή N κλειδιών. Σας δίνονται αρχεία κλειδιών με διαφορετικά N (50, 10₂, 10₃, 10₄, 10₅, 10₆). Πάρτε τις παρακάτω μετρήσεις για N=10₆ στο ΔΔΕ με array.

1. Μετρήστε το μέσο αριθμό συγκρίσεων ανά εισαγωγή.
2. Κάντε 100 αναζητήσεις τυχαίων αριθμών και μετρήστε τον μέσο αριθμό συγκρίσεων ανά αναζήτηση.
3. Κάντε 100 τυχαίες αναζητήσεις εύρους τιμών και μετρήστε τον μέσο αριθμό συγκρίσεων ανά αναζήτηση. Το εύρος ορίζεται με την αρχική τυχαία τιμή + K (όπου το K παίρνει τιμές 100 και 1.000)
4. Τον συνολικό χρόνο για N εισαγωγές.
5. Τον συνολικό χρόνο για 100 αναζητήσεις.

Πάρτε τις παραπάνω μετρήσεις σε μια κλάση ΔΔΕ (βρείτε μια έτοιμη υλοποίηση στο Web) με δυναμική παραχώρηση μνήμης που θα βρείτε από το Διαδίκτυο. Θα χρειαστεί να επέμβετε στην υλοποίηση ώστε να υπολογίζει τον αριθμό συγκρίσεων ανά πράξη εισαγωγής ή αναζήτησης και για να μετρήσετε τον χρόνο.

Κάντε μια διάσχιση inorder και για να κατασκευάσετε ένα ταξινομημένο πεδίο Nx1 με τις τιμές των κλειδιών. Πάρτε τις ίδιες μετρήσεις (εκτός από την περίπτωση εισαγωγής) για αναζήτηση στον ταξινομημένο πίνακα εφαρμόζοντας δυαδική αναζήτηση (binary search).

Συμπληρώστε τις τιμές στο παρακάτω πίνακα για N=10₆. Επίσης, φτιάξτε το διάγραμμα X-Y όπου Y είναι ο χρόνος εκτέλεσης και X είναι το N για τιμές (50, 10₂, 10₃, 10₄, 10₅, 10₆). Τα μαυρισμένα κελία στον πίνακα σημαίνουν ότι δεν θα πάρετε μετρήσεις αφού στη περίπτωση του ταξινομημένου πεδίου δεν θα κάνετε εισαγωγές (τα κλειδιά θα τα πάρει έτοιμα με διάσχιση στο δυαδικό δένδρο).

³ Συμβουλευτείτε την υλοποίηση της σελ. 16 της διάλεξης για συνδεδεμένες λίστες.

⁴ Η υλοποίηση της στοίβας γίνεται όπως και στην περίπτωση της συνδεδεμένης λίστας.

⁵ <https://www.tutorialspoint.com/compute-elapsed-time-in-seconds-in-java>

| Μέθοδος | Μέσος αριθμός συγκρίσεων / εισαγωγή | Συνολικός χρόνος για N εισαγωγές | Μέσος αριθμός συγκρίσεων / τυχαία αναζήτηση | Συνολικός χρόνος για N αναζητήσεις | Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=100) | Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=1.000) |
|-----------------------------------|-------------------------------------|----------------------------------|---|------------------------------------|---|---|
| ΔΔΕ με δυναμική παραχώρηση μνήμης | | | | | | |
| ΔΔΕ με array | | | | | | |
| Ταξινομημένο πεδίο | | | | | | |

Τεκμηρίωση των Αποτελεσμάτων (2 μονάδες)

Σχολιάστε (ξεχωριστό αρχείο κειμένου) την απόδοση όλων των μεθόδων και προσπαθήστε να δικαιολογήσετε την απόδοση κάθε μεθόδου. Προτείνετε πιθανές βελτιώσεις (δεν ζητείται να τις υλοποιήσετε). **Δώστε ιδιαίτερη βαρύτητα στην ερμηνεία των αποτελεσμάτων και μην δείξετε μόνο τις τιμές.**

Συστάσεις:

- Μην προχωρήσετε στην υλοποίηση πριν καταλάβετε καλά την υλοποίηση συνδεδεμένης λίστας με array που συζητήθηκε στο μάθημα.
- Δώστε προσοχή στην αρχικοποίηση της λίστας διαθέσιμων θέσεων. Θα πρέπει όλες οι θέσεις του πεδίου να βρίσκονται στην στοίβα των διαθέσιμων θέσεων. Η αρχικοποίηση μπορεί να γίνει με τον ίδιο ακριβώς τρόπο όπως στην συνδεδεμένη λίστα. Με τον ίδιο τρόπο επίσης υλοποιούνται οι μέθοδοι getnode και freenode (αντί για next χρησιμοποιείστε τον δείκτη right).
- Το πρόγραμμά σας πρέπει να δουλέψει για την είσοδο κλειδιών σε αρχείο που θα δώσουμε εμείς κατά την διόρθωση της άσκησης.
- Η άσκηση πρέπει να τυπώνει στην οθόνη τις τιμές του παραπάνω πίνακα για οποιοδήποτε αρχείο κλειδιών που θα δίνεται στην είσοδο.

Παραδοτέα:

Ένα συμπιεσμένο zip αρχείο που περιέχει:

- Μία έκθεση 2-3 σελίδες με τα αποτελέσματα που σας ζητούνται δηλαδή, έναν πίνακα για $N=10^6$ και το διάγραμμα X-Y που δείχνει πως εξαρτάται ο χρόνος εκτέλεσης από τον N. **Δώστε ιδιαίτερη βαρύτητα στην ερμηνεία των αποτελεσμάτων** και μην δείξετε μόνο τις τιμές.
- Μία έκθεση που περιγράφει σε 1-2 σελίδες πώς φτιάχτηκε ο κώδικας (δηλ. για κάθε ερώτημα ποια είναι η γενική ιδέα της λύσης σε 3-4 προτάσεις), υπάρχουν σαφείς οδηγίες μετάφρασης από compiler και εκτέλεσης, τι λάθη έχει (αν έχει, περιπτώσεις που δεν

δουλεύει το πρόγραμμα, ή περιπτώσεις που κάνει περισσότερα από όσα σας ζητεί η άσκηση, τι χρησιμοποιήσατε από έτοιμα προγράμματα ή πηγές πληροφόρησης. Υποδείξετε ακόμα και πηγές στο WWW όπως Wikipedia ή Stackoverflow (πλήρης διεύθυνση σχετικών σελίδων).

- Ο κώδικας περιέχει συνοπτικά σχόλια που εξηγούν την υλοποίηση. Προσθέστε σχόλια σε μορφή javadoc στην αρχή της κάθε κλάσης και κάθε μεθόδου. Επίσης javadoc σχόλια πριν από κάθε member variable των κλάσεων. Και όπου απαιτείται μέσα στον κώδικά σας.
- Εκτός των παραπάνω, οι ασκήσεις βαθμολογούνται με άριστα εφόσον:
 - Το zip είναι πλήρες.
 - Οι κώδικες περνούν από compiler και εκτελούνται κανονικά και σωστά σε windows ή Linux περιβάλλον (Προσοχή: Θα πρέπει να κάνετε χρήση σχετικών directory paths και όχι απόλυτων, τα οποία ισχύουν μόνο για τους υπολογιστές σας).
 - Ο κώδικάς σας δουλεύει για οποιαδήποτε τιμή παραμέτρων.
- Οι ασκήσεις υποβάλλονται ηλεκτρονικά στον ιστοχώρο του μαθήματος και όχι με e-mail.
- Οι αντιγραφές (ακόμη και μέρους της υλοποίησης) μηδενίζονται.