

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΡΧΕΙΩΝ

2^η άσκηση

Ημερομηνία παράδοσης: 2 Μαΐου 2023

Η άσκηση είναι ατομική

Η άσκηση έχει σκοπό της εξοικείωση με δομές που χειρίζονται δεδομένα δύο διαστάσεων της μορφής (x,y) . Και οι δύο δομές που θα φτιάξετε μπορεί να θεωρηθούν επεκτάσεις του δυαδικού δένδρου έρευνας σε δύο διαστάσεις (όπως είναι τα σημεία στο επίπεδο). Δηλαδή η αναζήτηση (όπως και η εισαγωγή) γίνονται ταυτόχρονα πάνω σε δύο τιμές κλειδιών αντί για μία (όπως είναι στο Δυαδικό Δένδρο Έρευνας). Οι εφαρμογές είναι πολλές με πιο συνηθισμένη την παράσταση γεωγραφικών πληροφοριών (πόλεις στον χάρτη). Μπορείτε να χρησιμοποιήσετε υλοποίηση με δυναμική παραχώρηση μνήμης (έχει πιο εύκολη υλοποίηση). Επίσης θα βρείτε πολλές αναφορές στο internet (όπως στην Wikipedia).

Μπορείτε να θεωρήσετε ότι τα δεδομένα είναι σημεία σε μια περιοχή διαστάσεων $N \times N$. Αυτό σημαίνει ότι $0 \leq x \leq N-1$ και $0 \leq y \leq N-1$. Τα βασικό σημείο της κάθε δομής (αντίστοιχα με το δυαδικό δένδρο έρευνας) είναι το κριτήριο διάσχισης δηλαδή πώς γίνεται η σύγκριση σε κάθε κόμβο προκειμένου να αποφασιστεί το μονοπάτι που ακολουθείται. Στην περίπτωση μας τα δεδομένα έχουν δύο πεδία x και y (που παριστάνουν τις συντεταγμένες ενός σημείου στο επίπεδο). Επομένως, η σύγκριση γίνεται σε δύο αντί για ένα κλειδί. Το πώς γίνεται εξαρτάται από την δομή και περιγράφεται παρακάτω. Η αναζήτηση για την εύρεση ενός κλειδιού γίνεται με διαδοχικές συγκρίσεις και μπορεί να φτάσει μέχρι τα φύλλα (οι δομές είναι δέντρα). Η εισαγωγή νέου κλειδιού προϋποθέτει αρχικά αναζήτηση και εφόσον το κλειδί δεν υπάρχει, εισαγωγή σε φύλλο.

Για κάθε δομή δεδομένων κατασκευάστε τις μεθόδους εισαγωγής και αναζήτησης στοιχείων.

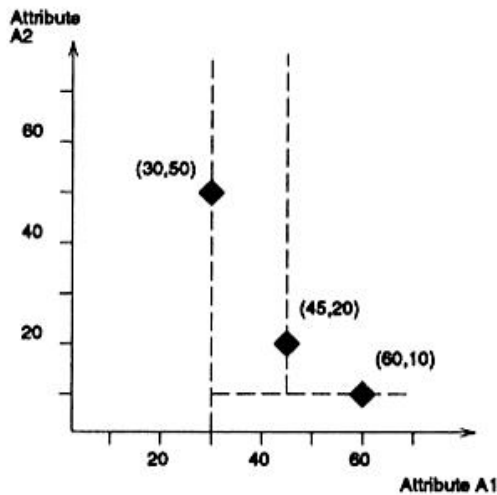
k-d Tree (25 μονάδες)

Υλοποιήστε την κλάση «k-d-tree». Το δέντρο είναι δυαδικό και μοιάζει με Δυαδικό Δένδρο Έρευνας (ΔΔΕ) με την διαφορά ότι κατά την διάσχιση του δέντρου η σύγκριση σε κάθε κόμβο γίνεται εναλλάξ πάνω στο x και στο y . Στο πρώτο επίπεδο (ρίζα) η σύγκριση γίνεται χρησιμοποιώντας την τιμή του x , στο επόμενο (δεύτερο) επίπεδο με την τιμή του y , στο τρίτο επίπεδο ξανά με το x , στο τέταρτο επίπεδο με το y και συνεχίζουμε το ίδιο.

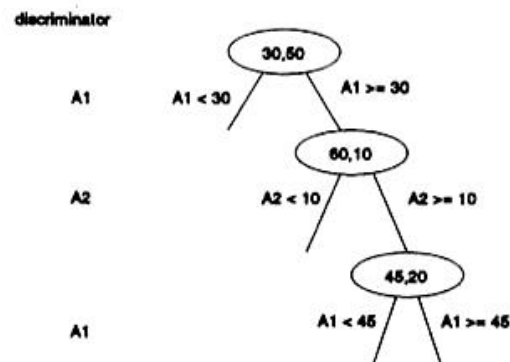
Κατά την εισαγωγή ενός στοιχείου (x_1, y_1) κάνουμε το εξής: Αρχικά, τοποθετούμε στην ρίζα του δέντρου το πρώτο στοιχείο (x_1, y_1) . Στο εξής, όλα τα στοιχεία με $x < x_1$ θα ανήκουν στο αριστερό υπο δέντρο του (x_1, y_1) και τα στοιχεία με $x \geq x_1$ στο δεξιό υποδέντρο. Μετά εισάγεται το επόμενο στοιχείο (x_2, y_2) και αποφασίζεται αν θα μπει στο αριστερό ή στο

δεξιό υποδένδρο του (x_1, y_1) . Το (x_2, y_2) θα μπει στην ρίζα του αριστερού υποδένδρου αν $x_2 < x_1$. Το (x_2, y_2) θα μπει στην ρίζα του δεξιού υποδένδρου αν $x_2 \geq x_1$. Για να βρούμε το σημείο εισαγωγής για το επόμενο στοιχείο (x_3, y_3) συγκρίνουμε πρώτα το x_3 με το x_1 και μετά το y_3 με το y του κόμβου που έχει επιλεγεί από την προηγούμενη σύγκριση. Αν η σύγκριση σε ένα επίπεδο γίνει με την τιμή x , στο επόμενο επίπεδο θα γίνει με την τιμή y . Συνεχίζουμε την διαδικασία μέχρι **το κάθε στοιχείο να βρεθεί σε ένα κόμβο του k-d-tree**. Δεν έχει τόσο μεγάλη σημασία αν θα εισαχθεί στοιχείο (x, y) που ήδη υπάρχει στο δέντρο. Για ευκολία υποθέτουμε ότι το στοιχείο εισάγεται όταν δεν υπάρχει.

Δείτε το παράδειγμα. Το σχήμα (α) δείχνει τα στοιχεία στο επίπεδο και το σχήμα (β) δείχνει το δέντρο που προκύπτει.



(a)



(b)

Στην αναζήτηση, όταν δίνεται η τιμή (x, y) και ψάχνουμε αν υπάρχει, αρχικά συγκρίνουμε το x με την x_1 συντεταγμένη της ρίζας. Αν βρούμε την τιμή (x, y) σταματάμε εκεί. Διαφορετικά, αν $x < x_1$, πηγαίνουμε στο αριστερό υποδέντρο, αλλιώς πάμε στο δεξιό. Στο επόμενο επίπεδο συνεχίζουμε το ψάξιμο ως προς την y . Αν δεν βρούμε το (x, y) συνεχίζουμε στο επόμενο επίπεδο συγκρίνοντας την x με τον ίδιο τρόπο και αν δεν βρούμε αυτό που ψάχνουμε συνεχίζουμε στο επόμενο επίπεδο με την y . Αυτό συνεχίζεται έως ότου βρούμε την τιμή (x, y) ή NULL.

PR Quadtree (35 μονάδες)

Σε δέντρο αυτού του τύπου, η $N \times N$ περιοχή αρχικά υποδιαιρείται σε 4 ίδιου μεγέθους υποπεριοχές, υποδιαιρώντας κάθε διάσταση με το 2. Η ρίζα του δέντρου, που αντιστοιχεί στην περιοχή $N \times N$, έχει δηλαδή 4 δείκτες προς 4 υποπεριοχές, που είναι τα φύλλα του δέντρου. Στοιχεία (x, y) **εισάγονται μόνο σε φύλλα** του δέντρου, που περικλείει το σημείο (x, y) . Κάθε φύλλο έχει το πολύ ένα μόνο στοιχείο.

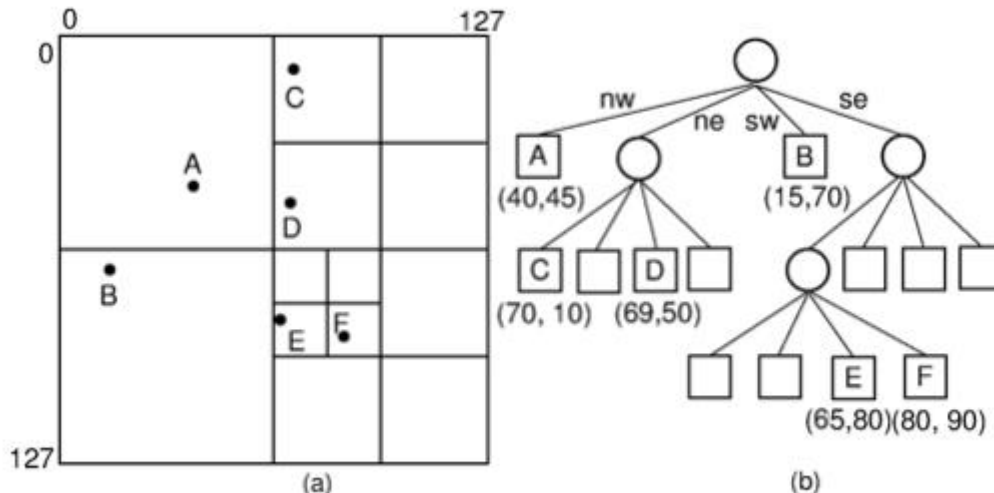
Αν το φύλλο στο οποίο θέλουμε να εισάγουμε ένα στοιχείο, περιέχει ήδη κάποιο άλλο στοιχείο, υποδιαιρούμε το φύλλο σε άλλες 4 υποπεριοχές και μετακινούμε το υπάρχον

στοιχείο στο σωστό νέο φύλλο. Εφόσον το νέο μας στοιχείο αντιστοιχεί σε φύλλο χωρίς άλλο στοιχείο, το εισάγουμε. Διαφορετικά, υποδιαιρούμε εκ νέου και επαναλαμβάνουμε μέχρι να προκύψει φύλλο χωρίς άλλο στοιχείο. Στο τέλος, κάθε φύλλο του δέντρου έχει το πολύ ένα στοιχείο.

Υποθέτουμε ότι ένα στοιχείο εισάγεται όταν δεν υπάρχει ήδη στο δέντρο (δεν έχουμε δηλαδή διπλά στοιχεία). Με αυτόν τον τρόπο προκύπτει ένα δέντρο όπου φύλλα μπορεί να περιέχουν ένα σημείο (x,y) , ενώ εσωτερικοί κόμβοι περιέχουν μόνο δείκτες σε 4 υποπεριοχές ίδιου μεγέθους.

Η αναζήτηση στοιχείου (x,y) ξεκινά από την ρίζα βρίσκοντας την περιοχή που περικλείει το στοιχείο (x,y) και συνεχίζουμε το ίδιο σε κάθε επόμενο επίπεδο του δέντρου μέχρι να φτάσουμε στα φύλλα.

Δείτε το παράδειγμα. Το σχήμα (α) δείχνει τα στοιχεία στο επίπεδο και το σχήμα (β) δείχνει το δέντρο που προκύπτει.



Εκτίμηση – Σύγκριση Απόδοσης (20 μονάδες αν κάνετε και τις δύο δομές δεδομένων ή 10 μονάδες για μόνο τη μία δομή δεδομένων)

Εισάγετε σε κάθε μια από τις παραπάνω δύο δομές **M** δεδομένα της μορφής (x,y) όπου τα **x**, **y** είναι τυχαίοι ακέραιοι αριθμοί στο **διάστημα** $N \times N$. Θεωρήστε ότι $N=2^{16}$. Κατασκευάστε τις δύο δομές δέντρου για διαφορετικό αριθμό δεδομένων. Κατασκευάστε κάθε δομή δεδομένων για **M = 200, 500, 1.000, 10.000, 30.000, 50.000, 70.000, 100.000**.

Στο κ-d-Tree και στο PR-Quatdtree (σε κάθε δομή δεδομένων χωριστά και για διαφορετικό αριθμό δεδομένων **M**, κάντε **100** αναζητήσεις για δεδομένα (x,y) που υπάρχουν και **100** αναζητήσεις για δεδομένα που δεν υπάρχουν. Για κάθε αναζήτηση (σε μια δομή δέντρου

και για κάθε **M**) υπολογίστε το βάθος του δέντρου στο οποίο φτάνουμε σε κάθε αναζήτηση.

Μετρήστε το μέσο όρο βάθους δέντρου που φτάνουμε σε κάθε περίπτωση επιτυχημένης και αποτυχημένης αναζήτησης. Η μέτρηση πρέπει να είναι με 2 δεκαδικά ψηφία.

Κατασκευάστε το διάγραμμα που δείχνει πώς μεταβάλλεται το βάθος που φτάνουμε για την εύρεση ενός στοιχείου ως συνάρτηση του **M** για κάθε μια από τις παραπάνω δύο περιπτώσεις αναζήτησης στις δύο δομές. Δηλαδή ο **Y** άξονας μετράει το μέσο βάθος δέντρου που φτάνουμε και ο **X** άξονας δείχνει το πλήθος δεδομένων στο δέντρο. Μπορείτε να χρησιμοποιήσετε κάποια εφαρμογή «gnuplot», «winplot» ή excel scatter chart κλπ για να φτιάξετε τα διαγράμματα.

Συνοπτικά: Κατασκευάστε ένα διάγραμμα με άξονες **X**, **Y** και σε αυτό το διάγραμμα θα παραστήσετε τις εξαρτήσεις:

- 1) k-d-Tree μέσος όρος βάθους δέντρου που φτάνουμε για επιτυχημένη αναζήτηση ως συνάρτηση του αριθμού στοιχείων **M**
- 2) k-d-Tree μέσος όρος βάθους δέντρου που φτάνουμε για αποτυχημένη αναζήτηση ως συνάρτηση του αριθμού στοιχείων **M**
- 3) PR Quadtree μέσος όρος βάθους δέντρου που φτάνουμε για επιτυχημένη αναζήτηση ως συνάρτηση του αριθμού στοιχείων **M**
- 4) PR Quadtree μέσος όρος βάθους δέντρου που φτάνουμε για αποτυχημένη αναζήτηση ως συνάρτηση του αριθμού στοιχείων **M**

Τεκμηρίωση Αποτελεσμάτων (10 μονάδες για το k-d, 10 για το Quad)

Σχολιάστε τα αποτελέσματα:

- 1) Να συγκρίνετε και να αιτιολογήσετε τις αποδόσεις επιτυχημένης και αποτυχημένης αναζήτησης κάθε δομής δεδομένων χωριστά.
- 2) Ποια έκφραση πολυπλοκότητας εκφράζει τις καμπύλες που σχεδιάσατε;
- 3) Ποια δομή είναι πιο γρήγορη και γιατί;
- 4) Δικαιολογήστε την απόδοση κάθε μεθόδου.

Προσέξτε ότι βαθμολογείται η απάντηση με βάση την τεκμηρίωση (δηλαδή είναι αναμενόμενο το αποτέλεσμα ή όχι και γιατί;)

Παραδοτέα: Ένα συμπίεσμένο zip αρχείο που περιέχει τον πηγαίο κώδικα (φάκελος src) και την ζητούμενη αναφορά (όχι zip μέσα στο zip!).

Γενικοί κανόνες για τον κώδικα και την αναφορά:

- Ο κώδικας περιέχει συνοπτικά σχόλια που εξηγούν την υλοποίηση.
- Μία έκθεση που περιγράφει σε 1-2 σελίδες πως 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.

- Οι ασκήσεις υποβάλλονται ηλεκτρονικά στον ιστοχώρο του μαθήματος και όχι με e-mail
- Οι αντιγραφές μηδενίζονται.
- Δώστε μεγάλη σημασία στην ποιότητα του κώδικα που θα στείλετε. Θα πρέπει να ακολουθεί τουλάχιστον τους παρακάτω κανόνες:
 - Χρησιμοποιείτε εύστοχα ονόματα μεταβλητών και μεθόδων έτσι ώστε να γίνεται εύκολα αντιληπτός ο λόγος και ο τρόπος χρήσης τους.
 - Γράψτε απλό και δομημένο κώδικα ώστε αυτός που τον διαβάζει να μπορεί να καταλάβει τα βήματα που ακολουθούνται για την υλοποίηση του κάθε προβλήματος.
 - Βάλτε εύστοχα σχόλια όπου χρειάζεται.
 - Κάντε το κώδικά σας παραμετρικό με χρήση μεταβλητών μέσω των οποίων θα μπορείτε να αλλάζετε κάποια μεγέθη (π.χ. τα μεγέθη για N, M κ.ο.κ.) χωρίς να χρειάζεται να αλλάζετε κάθε φορά και την αντίστοιχη υλοποίηση.