

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΕΡΓΑΣΙΑ 3

ΙΑΚΩΒΟΣ ΕΥΔΑΙΜΩΝ : 3130059

ΣΤΕΦΑΝΟΣ ΠΑΥΛΟΠΟΥΛΟΣ : 3130168

ΜΕΡΟΣ Α

Στο πρώτο μέρος της εργασίας υλοποιούμε δύο κλάσεις. Την `Rectangle()` και την `Point()`. Η `Point()` χρησιμοποιείται για να φτιάξουμε στιγμιότυπα τύπου σημείων σε ένα καρτεσιανό σύστημα με συντεταγμένες `x`, `y`. Μέσα από την `Point()` αναπτύσσουμε τις μεθόδους `int x()` που μας επιστρέφει την συντεταγμένη `x`. Την `int y()` που μας επιστρέφει την συντεταγμένη `y`. Την `double distanceTo(Point z)` που μας δίνει έναν αριθμό κινητής υποδιαστολής ο οποίος εκφράζει την απόσταση ενός σημείου από ένα άλλο. Την μέθοδο `int squareDistanceTo(Point z)` που μας δίνει έναν ακέραιο αριθμό ο οποίος εκφράζει την απόσταση ενός σημείου από ένα άλλο σημείο υψωμένη στο τετράγωνο και τέλος έχουμε την μέθοδο `String toString()` η οποία μια επιστρέφει ως σύνολο χαρακτήρων τις συντεταγμένες ενός σημείου με τη μορφή `(x,y)`. Έπειτα υλοποιούμε την κλάση `Rectangle()` η οποία δίνοντας της ο χρήστης ένα `x` ελάχιστο και `y` ελάχιστο και ένα `x` μέγιστο και `y` μέγιστο μας επιστρέφει ένα στιγμιότυπο ενός ορθογωνίου παραλληλογράμμου. Στην `Rectangle` συναντάμε τις μεθόδους `int xmin()` που μας επιστρέφει το ελάχιστο `X` του ορθογωνίου, `int ymin()` που μας επιστρέφει το ελάχιστο `y` του ορθογωνίου, `int xmax()` που μας επιστρέφει το μέγιστο `X` του ορθογωνίου, `int ymax()` που μας επιστρέφει το μέγιστο `y` του ορθογωνίου. Επίσης έχουμε τις μεθόδους `boolean contains(Point p)` που επιστρέφει `true` άμα ένα σημείο `p` βρίσκεται μέσα στο ορθογώνιο ή στα σύνορα του (πάνω στις πλευρές του) αλλιώς επιστρέφει `false`. Την `boolean intersects(Rectangle that)` που επιστρέφει `true` άμα δύο ορθογώνια έχουν κοινά σημεία αλλιώς επιστρέφει `false`. Την `double distanceTo(Point p)` που μας δίνει έναν αριθμό κινητής υποδιαστολής ο οποίος εκφράζει την απόσταση ενός σημείου από ένα ορθογώνιο. Δεν ενδιαφερόμαστε για την απόσταση άμα το σημείο περιλαμβάνεται μέσα στο ορθογώνιο. Τέλος έχουμε τις μεθόδους `int squareDistanceTo(Point p)` η οποία μας δίνει έναν ακέραιο αριθμό ο οποίος εκφράζει την απόσταση ενός σημείου από ένα ορθογώνιο υψωμένη στο

τετράγωνο και η `String toString()` η οποία μας επιστρέφει τις ελάχιστες και μέγιστες συντεταγμένες του `x` και `y` στην μορφή `[xmin,xmax] x [ymin,ymax]`.

ΜΕΡΟΣ Β

Στο δεύτερο μέρος της εργασίας κατασκευάζουμε ένα δέντρο 2-B με παρόμοια χαρακτηριστικά με το δυαδικό δέντρο και με κύρια διαφορά ότι το δέντρο μας σε κάθε κόμβο έχει δύο κλειδιά αφού κάθε κόμβος αποτελεί ένα σημείο του καρτεσιανού συστήματος όποτε έχει σαν κλειδιά τις συντεταγμένες `x` και `y` ενός σημείου. Η κλάση μας `TwoDTree` έχει τις εξής μεθόδους. Την `boolean isEmpty()` η οποία μας επιστρέφει `true` άμα το δέντρο μας είναι άδειο αλλιώς `false`. Την `int size()` που μας επιστρέφει τον αριθμό των κόμβων στο δέντρο μας, την `void insert(Point p)` η οποία εισάγει έναν κόμβο στο δέντρο μας. Την `boolean search(Point p)` η οποία επιστρέφει `true` άμα το σημείο `p` είναι ίδιο με κάποιο κόμβο του δέντρου μας, την `Point nearestNeighbor(Point p)` η οποία μας επιστρέφει τον κόμβο που βρίσκεται πιο κοντά σε ένα σημείο `p`. Τέλος, έχουμε την `List<Point> rangeSearch(Rectangle rect)` η οποία μας επιστρέφει μία λίστα με τους κόμβους του δέντρου που περιέχονται σε ένα δοσμένο στιγμιότυπο τύπου `Rectangle`. Επίσης κατασκευάσαμε και τρεις άλλες μεθόδους. Την `setSize()` η οποία ορίζει την τιμή του αριθμού κόμβων του δέντρου μας. Την μέθοδο `print()` η οποία εκτυπώνει τους κόμβους του δέντρου ανά επίπεδο και την μέθοδο `int HeightOfTree()` που μας δίνει το ύψος του δέντρου μας. Οι μέθοδοι `Point nearestNeighbor(Point p)` και `List<Point> rangeSearch(Rectangle rect)` για την υλοποίηση τους χρησιμοποιούν αντίστοιχα τις μεθόδους `squareDistanceTo` της κλάσης `Rectangle` αλλά και `Point` και την `intersects()` της κλάσης `Rectangle`. Στην κλάση `TwoDTree` έχουμε επίσης και την κλάση `TreeNode` που κατασκευάζει τους κόμβους του δέντρου καθώς και μια `main` μέθοδο η οποία διαβάζει ένα αρχείο κειμένου από το πληκτρολόγιο. Αφού το ανοίξει διαβάζει τα στοιχεία τοθ αρχείου σειρά-σειρά και δημιουργεί με αυτά αντικείμενα τύπου `Point()` τα οποία θα μπουν στο δέντρο μας. Έπειτα, ο χρήστης πρέπει να επιλέξει τρεις επιλογές. Την επιλογή νούμερο ένα όπου ο χρήστης δίνει ένα `query rectangle` δίνοντας από το πληκτρολόγιο τις διαστάσεις του. Την επιλογή νούμερο δύο όπου ο χρήστης δίνει ένα `query point` δίνοντας από το πληκτρολόγιο τις διαστάσεις του. Ή την επιλογή μηδέν για να τερματίσει το πρόγραμμα. Με την επιλογή της πρώτης επιλογής ουσιαστικά καλούμε την

μέθοδο `List<Point> rangeSearch(Rectangle rect)` , ενώ με την επιλογή
νούμερο δύο καλούμε την μέθοδο `nearestNeighbor(Point p)`.