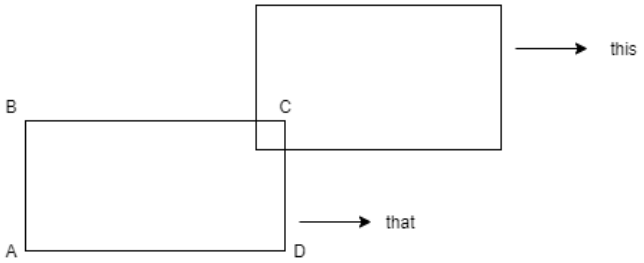
3η Εργασία

Οικονόμου Χρυσούλα|3170127

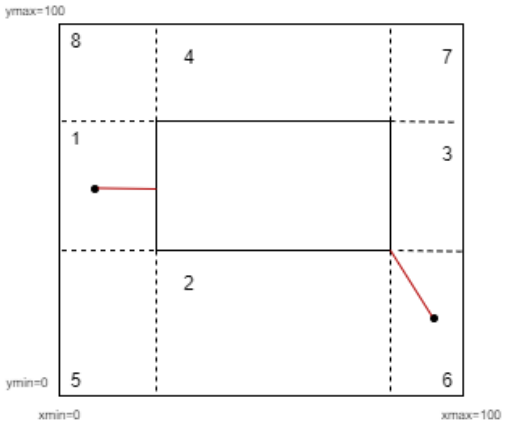
Ελευθέρου Μυρτώ-Χριστίνα|3170046

- contains(Point P): Εάν η συντεταγμένη x του σημείου που δίνεται, βρίσκεται ανάμεσα στο xmin και xmax του παραλληλογράμμου, και αντίστοιχα η συντεταγμένη y του σημείου p βρίσκεται ανάμεσα στο ymin και ymax, τότε το παραλληλόγραμμο περιέχει το σημείο p.
- intersects(Rectangle that): Για διευκόλυνση, ορίζουμε τα 4 σημεία του παραλληλόγραμμου που μας δίνεται χρησιμοποιώντας τα xmin, xmax, ymin, ymax. Εάν το παραλληλόγραμμο, περιέχει έστω ένα από τα 4 σημεία, τότε σημαίνει ότι τα 2 παραλληλόγραμμα «συναντιούνται».



- distanceTo(Point p): Εάν το σημείο περιέχεται στο παραλληλόγραμμο, τότε η συνάρτηση επιστρέφει 0. Διαφορετικά, ελέγχουμε που ακριβώς βρίσκεται το σημείο. Η απόσταση υπολογίζεται από το σημείο του παραλληλογράμμου που βρίσκεται στην ίδια ευθεία με το σημείο p. Αν το σημείο βρίσκεται αριστερά από το παραλληλόγραμμο και ανάμεσα στις τιμές xmin και xmax, τότε θα πάρουμε την απόσταση από το p και το σημείο του παραλληλογράμμου με συντεταγμένη y, ίδια με την συντεταγμένη y του σημείου p. Αυτό θα είναι το κοντινότερο σημείο του παραλληλογράμμου από το p. Ο έλεγχος αυτός γίνεται για όλα τα σημεία που βρίσκονται στις περιπτώσεις 1,2,3,4 και με αντίστοιχο τρόπο

υπολογίζουμε και τις συντεταγμένες του σημείου που θέλουμε να χρησιμοποιήσουμε για να βρούμε την απόσταση. Για τις υπόλοιπες περιπτώσεις (5,6,7,8), ελέγχουμε αν το σημείο βρίσκεται κάτω αριστερά/δεξιά ή πάνω αριστερά/δεξιά και χρησιμοποιούμε την αντίστοιχη γωνία του παραλληλογράμμου.



- nearestNeighbor(Point p):Εκτελώντας ένα while loop, διασχίζουμε όλα τα στοιχεία του δέντρου και τα προσθέτουμε σε μία λίστα. Στην συνέχεια, ξεκινώντας από το πρώτο στοιχείο της λίστας, μετράμε την απόσταση(distance). Με ένα for loop ελέγχουμε αν υπάρχει κάποιο σημείο το οποίο έχει μικρότερη απόσταση από την distance. Αν ναι, αποθηκεύουμε το σημείο αυτό και την απόστασή του. Τέλος επιστρέφουμε το σημείο με την μικρότερη απόσταση.
- rangeSearch(Rectangle rect):Εάν το head δεν είναι null, τότε καλούμε την αναδρομική συνάρτηση rangeSearchRecursive. Στην rangeSearchRecursive, ελέγχουμε αρχικά αν το παραλληλόγραμμο στο οποίο αντιστοιχεί ο τρέχων κόμβος «διασταυρώνεται» με το παραλληλόγραμμο rect, αν ναι, τότε ελέγχουμε αν το rect περιέχει το τρέχων σημείο και το προσθέτουμε στη λίστα. Στη συνέχεια καλούμε τη συνάρτηση rangeSearchRecursive για το δεξί και το αριστερό παιδί του κόμβου(αν υπάρχει). Τέλος η rangeSearchRecursive επιστρέφει την λίστα.