ΝΙΚΟΣ ΜΗΤΣΑΚΗΣ - Α.Μ. : 3210122
ΠΑΝΑΓΙΩΤΗΣ ΜΟΣΧΟΣ – Α.Μ. : 3210127

3η ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ ΤΩΝ ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ

ΜΕΡΟΣ Δ

Για το μέρος Α:

- Η μέθοδος contains της Rectangle παίρνει ως όρισμα ένα αντικείμενο τύπoy Point και ελέγχει με μία απλή λογική έκφραση αν οι συντεταγμένες x και y του σημείου αυτού είναι μεταξύ της τιμής της αντίστοιχης συντεταγμένης του πάνω δεξιά σημείου και του κάτω αριστερά σημείου (δηλαδή μεταξύ των τιμών x και y των δύο άκρων) του παραλληλογράμμου.
- Η μέθοδος distanceTo που παίρνει ως όρισμα ένα σημείο και υπολογίζει την ευκλείδεια απόσταση του σημείου αυτού από το παραλληλόγραμμο. Το κάνει αυτό υπολογίζοντας πρώτα την οριζόντια απόσταση (στον άξονα xx’ δηλαδή) παίρνοντας τη μέγιστη τιμή μεταξύ του απολύτου της διαφοράς των x συντεταγμένων του πάνω δεξιά σημείου του παραλληλογράμμου με τη x συντεταγμένη του δοσμένου σημείου και της αντίστοιχης διαφοράς του κάτω αριστερά σημείου του παραλληλογράμμου, πάλι με το δοσμένο σημείο. Η μέγιστη αυτή τιμή αποθηκεύεται σε μία μεταβλητή. Ομοίως, υπολογίζεται και η κατακόρυφη απόσταση (στον άξονα yy’ δηλαδή) πάλι ως η μέγιστη τιμή μεταξύ των απολύτων των διαφορών των y συντεταγμένων του πάνω δεξιά και του κάτω αριστερά με το δοσμένο σημείο. Η μέγιστη αυτή τιμή αποθηκεύεται σε μία άλλη μεταβλητή. Τέλος, η μέθοδος επιστρέφει την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των δύο μεταβλητών. (Σύμφωνα με το Πυθαγόρειο Θεώρημα)
- Συμπληρωματικά, για να αποφύγουμε την έλλειψη ακρίβειας μου ενδέχεται να συμπεριλάβει στις πράξεις ο υπολογισμός τετραγωνικών ριζών, υλοποιήσαμε και την μέθοδο squareDistanceTo η οποία υπολογίζει το

τετράγωνο της απόστασης ενός σημείου από ένα παραλληλόγραμμο.

Για το μέρος Β:

- Η μέθοδος rangeSearch της κλάσης TwoDTree, η οποία επιστρέφει μία απλά συνδεδεμένη λίστα (από την 1η εργασία - ***απαιτείται μία φορά compile με utf-8 encoding στο SinglyLinkedList.java ή στο TwoDTree.java πρώτα***) που περιέχει όλα τα σημεία (αντικείμενα τύπου Point) που ανήκουν στο δέντρο και βρίσκονται εντός του δοσμένου παραλληλογράμμου. Για την υλοποίησή της γράψαμε και μία βοηθητική μέθοδο, με όνομα rangeSearchHelper η οποία καλείται εσωτερικά της rangeSearch και κάνει την «πολύ» δουλειά. Συγκεκριμένα η rangeSearchHelper δέχεται ως ορίσματα ένα αντικείμενο TreeNode, ένα αντικείμενο Rectangle και μία απλά συνδεδεμένη λίστα. Η μέθοδος (όπως και οι περισσότερες σε αυτή την κλάση) δουλεύει αναδρομικά. Δηλαδή, εφόσον ο δοσμένος κόμβος είναι έγκυρος (δηλαδή δεν είναι null), ελέγχουμε αν το παραλληλόγραμμο που αντιστοιχεί στον κόμβο αυτό «κόβει» (χρησιμοποιώντας την intersects) το δοσμένο παραλληλόγραμμο και έπειτα εφόσον αυτό ισχύει ελέγχουμε αν το σημείο που είναι αποθηκευμένο στον συγκεκριμένο κόμβο περιέχεται (χρησιμοποιώντας την contains) μέσα στο δοσμένο παραλληλόγραμμο. Αν πληρούνται αυτές οι συνθήκες τότε προσθέτουμε το σημείο του τρέχοντος κόμβου στην λίστα μας. Αν όχι, αναδρομικά ψάχνουμε τα αριστερά και τα δεξιά υποδέντρα του τρέχοντος κόμβου, συγκεκριμένα ψάχνοντας πρώτα αριστερά και μετά δεξιά. Τέλος, η μέθοδος rangeSearch που δέχεται ως όρισμα ένα

αντικείμενο TreeNode (το οποίο οφείλει να είναι η ρίζα του δέντρου ώστε η διάσχιση του δέντρου να αρχίσει από την αρχή του) και ένα αντικείμενο Rectangle, δημιουργεί μία απλά συνδεδεμένη λίστα, ώστε να καλέσει την rangeSearchHelper με αυτά τα τρία ορίσματα και τέλος επιστρέφει την επεξεργασμένη πλέον λίστα με τα κατάλληλα σημεία.

- Την μέθοδο nearestNeighbor της κλάσης TwoDTree η οποία δέχεται ως όρισμα ένα σημείο που να ανήκει στην περιοχή [0, 100]x[0, 100] και επιστρέφει ένα σημείο που είναι μέσα στο δέντρο από το οποίο απέχει την ελάχιστη απόσταση, την υλοποιήσαμε ως εξής. Αρχικά εκτελούμε τους απαραίτητους ελέγχους, πρώτιστα για το αν το δέντρο μας είναι άδειο, οπότε και επιστρέφουμε null και εκτυπώνουμε κατάλληλο μήνυμα στον χρήστη και στη συνέχεια ελέγχουμε την περίπτωση το δοσμένο σημείο να βρίσκεται μέσα στο δέντρο, οπότε και επιστρέφουμε το ίδιο το σημείο, εκτυπώνοντας βέβαια το κατάλληλο μήνυμα. Εφόσον δεν Δημιουργούμε μία απλά συνδεδεμένη λίστα στην οποία αποθηκεύουμε όλα τα σημεία που μας επιστρέφει η rangeSearch όταν την καλέσουμε με ορίσματα την ρίζα του δέντρου και ως ορθογώνιο το [0, 100]x[0, 100]. Με αυτό τον τρόπο θα έχουμε όλα τα σημεία που ανήκουν στο δέντρο, καθώς θα βρίσκονται απαραίτητα στο επίπεδο αυτό. Επομένως, με ένα απλό while loop μπορούμε στη συνέχεια να ελέγξουμε ποιο από τα σημεία του δέντρου απέχει την ελάχιστη απόσταση από το δοσμένο σημείο, την οποία και εκτυπώνουμε, ενώ επιστρέφουμε το κοντινότερο σημείο που βρήκαμε.
- Για δική μας διευκόλυνση, υλοποιήσαμε με απλό τρόπο και μία μέθοδο toString στη κλάση του δέντρου, η οποία τυπώνει όλους τους κόμβους/σημεία που υπάρχουν

μέσα του ακολουθώντας το μοτίβο της προδιατεταγμένης διάσχισης συγκεκριμένα.