ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ
Φθινοπωρινό εξάμηνο 2022-2023
3η Εργασία

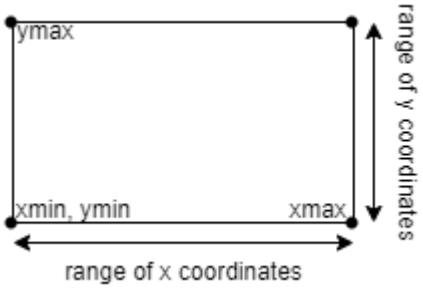
Ονοματεπώνυμο:	Αριθμός Μητρώου:
Γεώργιος Βεάζογλου	3190347
Βασιλική Δρόσου	3200044

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Μέρος Α

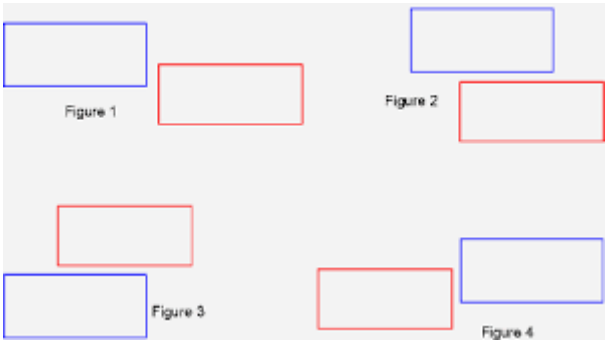
Μέθοδος boolean contains(Point p):

Έστω ένα ορθογώνιο που ορίζεται από ένα σύνολο σημείων xmin, xmax, ymin, ymax. Εάν κάποιο άλλο σημείο p έχει συντεταγμένες (x, y) με τιμές x ∈ (range of x coordinates) και y ∈ (range of y coordinates), σημαίνει ότι το σημείο αυτό είτε θα είναι στα όρια του ορθογωνίου είτε θα περιέχεται στο ορθογώνιο. Επομένως η μέθοδος contains(Point p) θα επιστρέψει true. Αλλιώς θα επιστρέψει false



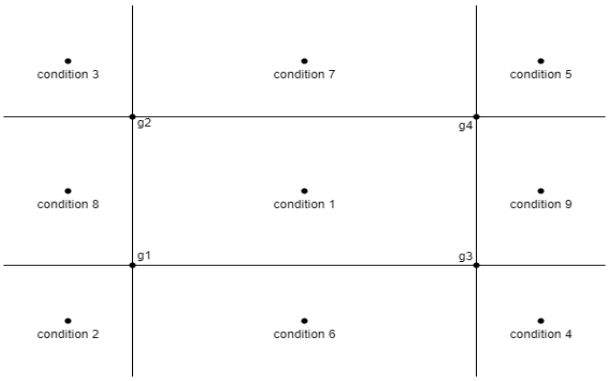
Μέθοδος boolean intersects(Rectangle that):

Στο διπλανό σχήμα, παρουσιάζονται όλα τα πιθανά σενάρια δυο ορθογωνίων που δεν διατέμνονται. Παρατηρούμε ότι για να συμβεί αυτό, θα πρέπει το ένα να είναι πάνω από το άλλο ή το ένα να βρίσκεται αριστερά του άλλου. Έτσι, χρησιμοποιώντας τις συντεταγμένες τους, υπολογίζουμε το ύψος και το πλάτος του καθενός και, στη συνέχεια, συγκρίνουμε τις τιμές με δύο συνθήκες if για να καθορίσουμε αν οποιαδήποτε από τις προαναφερθείσες προϋποθέσεις ισχύουν. Η συνάρτηση επιστρέφει true εάν δεν ισχύουν και false εάν κάποια από αυτές ισχύει.



Μέθοδος double distanceTo(Point p):

Στην αρχή της κλάσης ορίζονται τα σημεία g1, g2, g3, g4 που αναπαριστούν τις γωνίες του ορθογωνίων. Όπως αναπαρίσταται και στο διπλανό σχήμα, ένα σημείο p μπορεί να βρίσκεται είτε στο εσωτερικό (condition 1) είτε στο εξωτερικό ενός ορθογωνίου. Εάν βρίσκεται στο εξωτερικό του, διακρίνουμε 8 διαφορετικές περιπτώσεις (conditions 2-9). Έτσι, χρησιμοποιώντας προσαρμοσμένες συνθήκες if για κάθε περίπτωση, υπολογίζουμε την απόσταση του εκάστοτε σημείου p είτε συγκρίνοντας τις τιμές των συντεταγμένων (conditions 6-9), είτε καλώντας τη μέθοδο distanceTo(Point z) της κλάσης Point που υπολογίζει την ευκλείδεια απόσταση μεταξύ δύο σημείων (στην προκειμένη περίπτωση την απόσταση των σημείων p και g1-4.) (conditions 2-5).



Μέρος Β:

Μέθοδος public List<Point> rangeSearch(Rectangle rect):

Η μέθοδος rangeSearch λαμβάνει ένα αντικείμενο Rectangle ως είσοδο και επιστρέφει ένα αντικείμενο List<Point> που περιέχει όλα τα σημεία που περιέχονται μέσα στο ορθογώνιο. Ωστόσο, ο αλγόριθμος rangeSearch υλοποιείται στην overloaded μέθοδο rangeSearch(Rectangle rect, TreeNode node, List<Point> list).

Μέθοδος private void rangeSearch(Rectangle rect, TreeNode node, List<Point> list):

Η μέθοδος rangeSearch είναι μια ιδιωτική μέθοδος βοηθητικής λειτουργίας που διασχίζει αναδρομικά το δέντρο και προσθέτει σημεία που περιλαμβάνονται μέσα στο ορθογώνιο στη λίστα αποτελεσμάτων. Πρώτα, αν ο τρέχων κόμβος είναι κενός (null), η μέθοδος επιστρέφει χωρίς να κάνει τίποτα. Στη συνέχεια, εάν ο κόμβος δεν είναι κενός, δημιουργείται ένα αντικείμενο Rectangle για τον τρέχοντα κόμβο χρησιμοποιώντας τις συντεταγμένες x και y του περιέχοντος σημείου, και ελέγχεται εάν διατέμνεται με το ορθογώνιο που παρέχεται από τον χρήστη. Ο έλεγχος πραγματοποιείται με τη χρήση της μεθόδου intersects που υλοποιήθηκε στην κλάση Rectangle.Εάν δεν διατέμνεται, η μέθοδος επιστρέφει χωρίς να αναζητήσει τον τρέχον κόμβο ή τα υποδέντρα του. Εάν το σημείο του κόμβου περιλαμβάνεται στο ορθογώνιο, προστίθεται στη λίστα αποτελεσμάτων. Τέλος, η μέθοδος καλεί αναδρομικά τον εαυτό της στα αριστερά και δεξιά υποδέντρα.

Τέλος, να σημειωθεί ότι για να εκτελεστεί σωστά ο αλγόριθμος, στη μέθοδο rangeSearch(Rectangle rect), όταν κληθεί η βοηθητική της, περνάμε σαν argument στο TreeNode node τη ρίζα (head) του δέντρου.

Μέθοδος public Point nearestNeighbor(Point p):

Η μέθοδος nearestNeighbor λαμβάνει ένα αντικείμενο Point ως είσοδο, ελέγχει εάν η ρίζα (head) είναι null (αν αληθεύει η μέθοδος επιστρέφει null) και επιστρέφει ένα αντικείμενο Point. Ο αλγόριθμος nearestNeighbor υλοποιείται στην overloaded μέθοδο nearestNeighbor(TreeNode node, Point target, point closest).

Μέθοδος public static Point nearestNeighbor(TreeNode node, Point target, Point closest):

Η συνάρτηση αναδρομικά διασχίζει το δέντρο ξεκινώντας από τη ρίζα (head) και ελέγχει εάν ο τρέχων κόμβος είναι πιο κοντά στο σημείο-στόχο από το πιο κοντινό σημείο που έχει βρεθεί μέχρι στιγμής. Εάν ναι, το πλησιέστερο σημείο ενημερώνεται για να είναι το σημείο που αντιστοιχεί στον τρέχοντα κόμβο.Ο κώδικας ελέγχει αρχικά εάν το όρισμα κόμβου (node) είναι null. Εάν είναι, η συνάρτηση επιστρέφει το πιο κοντινό σημείο που έχει βρεθεί μέχρι στιγμής.Εάν το όρισμα κόμβου δεν είναι null, η συνάρτηση υπολογίζει τις αποστάσεις μεταξύ του σημείου του κόμβου (node.getPoint()) και του σημείου-στόχου (target), καθώς και μεταξύ του πιο κοντινού σημείου που έχει βρεθεί μέχρι στιγμής (closest) και του target. Εάν το σημείο του κόμβου είναι πιο κοντά στο target από το πιο κοντινό σημείο που έχει βρεθεί μέχρι στιγμής, το πλησιέστερο σημείο ενημερώνεται για να είναι το σημείο του κόμβου. Στη συνέχεια, η συνάρτηση ελέγχει εάν το target μπορεί να βρίσκεται στην άλλη πλευρά του επιπέδου διαχωρισμού του τρέχοντος κόμβου συγκρίνοντας την απόσταση μεταξύ του target και των ορθογωνίων των παιδικών κόμβων (εάν υπάρχουν) με την απόσταση μεταξύ του target και του πλησιέστερου σημείου που έχει βρεθεί μέχρι στιγμής. Εάν η απόσταση προς το ορθογώνιο ενός παιδικού κόμβου είναι μικρότερη από την απόσταση προς το πλησιέστερο σημείο, η συνάρτηση καλεί αναδρομικά τον εαυτό της σε

αυτόν τον παιδικό κόμβο για να συνεχίσει την αναζήτηση ενός πιο κοντινού σημείου. Τέλος, η συνάρτηση επιστρέφει το πλησιέστερο σημείο που βρέθηκε στο δέντρο.