

Αναφορά παράδοσης

Παπαθανασίου Αθανασία Μαρία : 3180147

Μέρος Α

Contains: Ένα σημείο περιέχεται στο παραλληλόγραμμο αν η τετμημένη του σημείου είναι μικρότερη ή ίση με τη μέγιστη τιμή της συντεταγμένης x του παραλληλογράμμου και η τεταγμένη του σημείου είναι μικρότερη ή ίση από τη μέγιστη τιμή της συντεταγμένης y του παραλληλογράμμου. Εάν ισχύει το παραπάνω τότε η συνάρτηση επιστρέφει true αλλιώς false.

Intersects: Η συνάρτηση επιστρέφει true αν τα παραλληλόγραμμα έχουν κοινά σημεία αλλιώς false. Οι περιπτώσεις όπου δεν έχουν κανένα κοινό σημείο είναι όταν το ένα είναι αριστερά του άλλου ή το ένα είναι πάνω από το άλλο. Για να ελέγξουμε την πρώτη περίπτωση βλέπουμε εάν η μικρότερη τιμή της συντεταγμένης x του ενός παραλληλογράμμου είναι μεγαλύτερη από τη μέγιστη συντεταγμένη x του άλλου παραλληλογράμμου και το αντίστροφο. Για να ελέγξουμε εάν το ένα παραλληλόγραμμο είναι πάνω από το άλλο πρέπει η μικρότερη τιμή της συντεταγμένης y του ενός παραλληλογράμμου να είναι μεγαλύτερη από τη μέγιστη συντεταγμένη y του άλλου παραλληλογράμμου και το αντίστροφο. Και στις δύο παραπάνω περιπτώσεις η συνάρτηση επιστρέφει false

distanceTo: Η συνάρτηση αυτή επιστρέφει την απόσταση του σημείου p από το κοντινότερο σημείο του παραλληλογράμμου. Συγκεκριμένα η απόσταση υπολογίζεται ως η τετραγωνική ρίζα της διαφοράς των τετμημένων στο τετράγωνο συν τη διαφορά των τεταγμένων στο τετράγωνο. Η συνάρτηση αυτή επιστρέφει double αλλά επειδή χρησιμοποιούμε την sqrt είναι πιθανόν να έχουμε αποκοπή κάποιων δεκαδικών ψηφίων.

Μέρος Β

rangeSearch: Η μέθοδος επιστρέφει μία λίστα με τα σημεία που περιέχονται στο παραλληλόγραμμο χρησιμοποιώντας μία βοηθητική συνάρτηση με το ίδιο όνομα. Αρχικά δημιουργούμε μία κενή λίστα και αρχικοποιούμε μία μεταβλητή boolean div η οποία καθορίζει ποια συντεταγμένη θα χρησιμοποιήσουμε σε κάθε επίπεδο. Στη συνέχεια δημιουργούμε ένα παραλληλόγραμμο που αντιπροσωπεύει τη ρίζα και είναι της μορφής $[0,100] \times [0,100]$ δηλαδή περιέχει όλα τα σημεία που μας αφορούν. Έπειτα καλούμε τη συνάρτηση με ορίσματα την ρίζα, τα δύο παραλληλόγραμμα, την div και τη λίστα. Η συνάρτηση αφού καλέσει τη βοηθητική τυπώνει τη λίστα και την επιστρέφει. Η βοηθητική συνάρτηση, χρησιμοποιεί αναδρομή προκειμένου να εντοπίσει τα σημεία που περιέχονται σε ένα παραλληλόγραμμο, ξεκινώντας από τη ρίζα. Συγκεκριμένα αν η ρίζα είναι null η συνάρτηση δεν επιστρέφει τίποτα. Έπειτα βλέπουμε αν τα δύο παραλληλόγραμμα έχουν κοινά σημεία, με χρήση της intersects, εάν έχουν ελέγχουμε με την contains αν το σημείο που βρήκαμε

περιέχεται στο παραλληλόγραμμα και αν ναι το τοποθετούμε στη λίστα χρησιμοποιώντας την `insertAtBack`. Εάν το `div` είναι `true` τότε χρησιμοποιούμε την `χ` συντεταγμένη και δημιουργούμε δύο παραλληλόγραμμα χρησιμοποιώντας τη συντεταγμένη `χ`. Συγκεκριμένα, τα σημεία του παραλληλογράμμου του αριστερού υπόδεντρου έχουν μικρότερη ή ίση συντεταγμένη `χ` με αυτή της ρίζας ενώ του δεξιού μεγαλύτερη, οπότε αναδρομικά καλούμε την συνάρτηση για το δεξί και το αριστερό υπόδεντρο αλλάζοντας τη μεταβλητή `div` σε `true` για να μπορέσει να κληθεί αναδρομικά η συνάρτηση χρησιμοποιώντας τη συντεταγμένη `γ`. Δηλαδή στο επόμενο επίπεδο του δέντρου το παιδί που βρίσκεται στο αριστερό υπόδεντρο πρέπει να έχει συντεταγμένη `γ` μικρότερη από αυτή του γονιού του, ενώ το αντίθετο ισχύει για το δεξί παιδί.

nearestNeighbor: Η μέθοδος αυτή χρησιμοποιεί επίσης βοηθητική συνάρτηση με το ίδιο όνομα την οποία καλεί αφού πρώτα δημιουργήσει ένα παραλληλόγραμμα που αντιπροσωπεύει τη ρίζα. Η βοηθητική συνάρτηση παίρνει ως όρισμα τη ρίζα του δέντρου, ένα σημείο `ρ`, το παραλληλόγραμμα που αντιπροσωπεύει τη ρίζα και το σημείο του δέντρου που είναι πιο κοντά στο `ρ` το οποίο αρχικά είναι `null`. Η συνάρτηση επιστρέφει το σημείο που είναι πιο κοντά στο `ρ` το οποίο ονομάζουμε `close` και στην `main` εκτυπώνουμε και την απόσταση μεταξύ των δύο αυτών σημείων. Ορίζουμε την απόσταση του σημείου `ρ` και του παραλληλογράμμου, μία μεταβλητή `div` η οποία αν είναι `true` χρησιμοποιούμε την συντεταγμένη `χ` αλλιώς την `γ` και αρχικοποιούμε το δεξί και αριστερό παραλληλόγραμμα ενός κόμβου. Αν η ρίζα είναι `null` τότε επιστρέφουμε `null`. Στη συνέχεια, αν το `close` έχει τιμή ορίζουμε την απόσταση των δύο σημείων και την απόσταση του παραλληλογράμμου από το `ρ`. Χρησιμοποιούμε το τετράγωνο της απόστασης για να μην έχουμε απώλεια ακρίβειας. Αν το τρέχον κοντινότερο σημείο που έχουμε βρει μέχρι εκείνη τη στιγμή έχει απόσταση από το `ρ` αυστηρά μικρότερη από την απόσταση του `ρ` από το παραλληλόγραμμα στο οποίο αντιστοιχεί κάποιος κόμβος τότε πάλι δεν χρειάζεται να ψάξουμε αυτόν τον κόμβο και τα υποδέντρα του και απλά επιστρέφουμε τον κοντινότερο κόμβο στο `ρ`. Βάζουμε σε ένα `ρ2` τις συντεταγμένες του τρέχοντος σημείου, το οποίο πλέον θα ισούται με το κοντινότερο σημείο του `ρ`. Αν το `div` είναι `true` αλλάζουμε την τιμή του σε `false` και χρησιμοποιούμε τη συντεταγμένη `χ` για αυτό το επίπεδο. Δημιουργούμε δύο παραλληλόγραμμα με την ίδια λογική που τα δημιουργήσαμε και στην `rangeSearch`. Συγκεκριμένα, τα σημεία του παραλληλογράμμου του αριστερού υπόδεντρου έχουν μικρότερη ή ίση συντεταγμένη `χ` με αυτή της ρίζας ενώ του δεξιού μεγαλύτερη, οπότε αναδρομικά καλούμε την συνάρτηση για το δεξί και το αριστερό υπόδεντρο αλλάζοντας τη μεταβλητή `div` σε `true` για να μπορέσει να κληθεί αναδρομικά η συνάρτηση χρησιμοποιώντας τη συντεταγμένη `γ`. Οπότε με βάση αυτό, αν η τετμημένη του σημείου `ρ` είναι μικρότερη από την τετμημένη του κόμβου τότε καλούμε αναδρομικά τη συνάρτηση πρώτα για το αριστερό και έπειτα για το δεξί υπόδεντρο και βάζουμε την τιμή της στο `close`. Αν η τετμημένη του σημείου `ρ` είναι μεγαλύτερη ή ίση από την τετμημένη του κόμβου τότε καλούμε αναδρομικά τη συνάρτηση πρώτα για το δεξί και έπειτα για το αριστερό

υπόδεντρο. Αντίστοιχα, αν το `div` είναι `false` συγκρίνουμε τις συντεταγμένες y των σημείων. Τέλος επιστρέφουμε το `close` στην βασική συνάρτηση και στην `main` εκτυπώνουμε και την απόσταση μεταξύ των δύο σημείων.