

Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής
Μάθημα: Δομές Δεδομένων
Ακαδημαϊκό έτος: 2019–20
Μέλη: Ελένη Σαζώνη-3160153, Αριστείδης Χρονόπουλος-3160194

Εργασία 3

Μέρος Α:

Ξεκινώντας από την κλάση `Rectangle`, η οποία δέχεται ως ορίσματα `xmin`, `xmax`, `ymin`, `ymax`.

contains(Point p): Η `contains` δέχεται ως όρισμα ένα σημείο `p` τύπου `point` με `x` και `y` συντεταγμένες και επιστρέφει `true` αν το σημείο ανήκει στο παραλληλόγραμμο και `false` στην αντίθετη περίπτωση. Για να το υλοποιήσουμε αυτό ελέγχουμε εάν το σημείο `x` του `p` βρίσκεται ανάμεσα στο `xmin` και `xmax` του παραλληλόγραμμου και εάν το `y` είναι ανάμεσα στο `ymin` και `ymax` του παραλληλογράμμου.

intersects(Rectangle t): Με την μέθοδο `intersects` θέλουμε να διαπιστώσουμε εάν το παραλληλόγραμμο του ορίσματος συμπίπτει με το αντικείμενο μας. Αυτό το υλοποιούμε συγκρίνοντας τα `xmin` και τα `xmax` του καθενός.

```
if(that.xmin <= this.xmax || this.xmin <= that.xmax)
{
    return true;
}else{
    return false; }
```

distanceTo(Point p): Βρίσκοντας την ευκλείδεια απόσταση από το σημείο `p` από κάθε μια πλευρά του παραλληλογράμμου και διαλέγοντας κάθε φορά την μικρότερη. _

Μέρος Β:

- **Range search:**

Με την range search βρίσκουμε πόσα από τα σημεία ενός δυαδικού δέντρου βρίσκονται μέσα σε ένα παραλληλόγραμμο που δίνεται από τον χρήστη. Για την υλοποίηση της `rangeSearch` χρησιμοποιήσαμε μια βοηθητική μέθοδο, την `range`. Το σκεπτικό που χρησιμοποιήσαμε ήταν ότι κάθε φορά κρατάμε τον current Treenode καθώς και το παραλληλόγραμμο του, το οποίο κάθε φορά βρίσκουμε χρησιμοποιώντας την `range`. Με την `range` κάθε φορά ελέγχουμε αν δουλεύουμε με τα x ή με τα y του parent node και προσδιορίζουμε το Rectangle του current. Οπότε κάθε φορά ελέγχουμε ταυτόχρονα τα παραλληλόγραμμα των παιδιών που προσδιορίζουμε αν συμπίπτουν με το δεδομένο παραλληλόγραμμο (intersects) προχωράμε στο συγκεκριμένο υποδέντρο χρησιμοποιώντας την ίδια διαδικασία. Κάθε φορά ελέγχουμε αν ο parent node βρίσκεται στο παραλληλόγραμμο που θέλουμε και τον προσθέτουμε στην λίστα.

- **Nearest Neighbor Search:**

Για την συγκεκριμένη μέθοδο ξεκινάμε από την ρίζα του δέντρου, αρχικοποιώντας στο `min_p` (το στοιχείο που απέχει τη λιγότερη απόσταση), την ρίζα. Διασχίζοντας το δέντρο, αν βρούμε σημείο που απέχει λιγότερη απόσταση από το `min_p`, τα ανταλλάζουμε. Για να διασχίσουμε το δέντρο πρέπει να ξέρουμε αν θα πάμε στο αριστερό ή στο δεξί υποδέντρο. Για να το καταφέρουμε αυτό, συγκρίνουμε ποια από τις ρίζες των υποδέντρων έχει την μικρότερη `distanceTo`, και προχωράμε στο συγκεκριμένο. Για περαιτέρω υλοποίηση της συγκεκριμένης μεθόδου χρησιμοποιήσαμε τις `findLeftRect` και `findRightRect`.

Για να τρέξετε την κλάση `TwoDTree`:

```
> java TwoDTree inputdata.txt
```