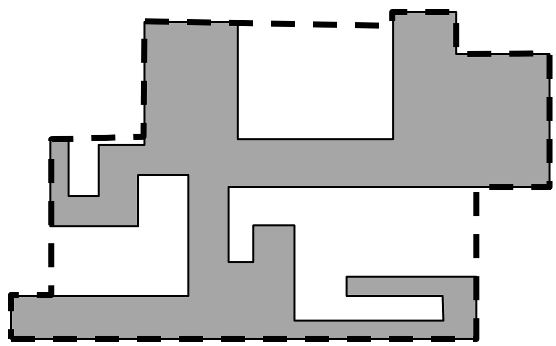
Ασκήσεις στην Υπολογιστική Γεωμετρία

Βαθμολογικό βάρος 7

Καταληκτική Ημ/νια Παράδοσης: ??/6/2016 (με email στο tsichlas@delab.csd.auth.gr)

Παραδοτέο θα είναι μία αναφορά με τις απαντήσεις καθώς και ο κώδικας όπου σας ζητείται (και το εκτελέσιμο αρχείο βεβαίως). Η εργασία μπορεί να γίνει και σε ομάδες των δύο ατόμων.

Άσκηση 1: (0.5) Ένα απλό πολύγωνο λέγεται ορθογώνιο αν κάθε ακμή του είναι παράλληλη είτε στον x-άξονα ή στον y-άξονα. Ένα ορθογώνιο πολύγωνο είναι ορθογώνια κυρτό αν η τομή του με οποιαδήποτε οριζόντια ή κάθετη γραμμή είναι ένα συνεκτικό σύνολο (τέμνει σε δύο ή καμία ακμές του ορθογωνίου). Το ορθογώνιο κυρτό περίβλημα ενός ορθογωνίου πολυγώνου P είναι το μικρότερο ορθογώνιο κυρτό πολύγωνο που καλύπτει το P. Στο παρακάτω σχήμα με την διακεκομμένη γραμμή συμβολίζουμε το ορθογώνιο περίβλημα του P, το οποίο είναι σκιασμένο.

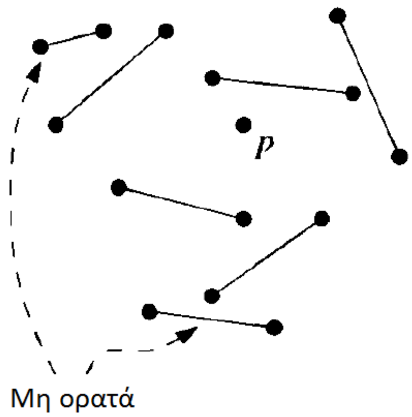


Σας ζητούνται τα εξής:

1. Περιγράψτε ένα ταχύ αλγόριθμο για τον υπολογισμό ενός ορθογωνίου κυρτού περιβλήματος ενός ορθογωνίου πολυγώνου που δίνεται ως είσοδο. Ποια είναι η πολυπλοκότητα του αλγορίθμου σας σε σχέση με το πλήθος των κορυφών του ορθογωνίου πολυγώνου που μας δίνεται ως είσοδο (με απόδειξη);
2. **(BONUS – 1 μονάδα)** Στην ιστοσελίδα του βιβλίου Computational Geometry in C ([1] στην ιστοσελίδα του μαθήματος από όπου μπορείτε να βρείτε και την ιστοσελίδα του βιβλίου) υπάρχει κώδικας σε Java ενός applet με υλοποιημένους κάποιους αλγόριθμους. Θα πρέπει να υλοποιήσετε τον αλγόριθμο που προτείνατε στο (1) σε αυτό το γραφικό περιβάλλον. Επιπλέον, θα πρέπει να «πειράξετε» το περιβάλλον ώστε να μπορείτε να σχεδιάσετε ορθογώνια πολύγωνα. Το παραδοτέο θα είναι όλος ο κώδικας με το αντίστοιχο εκτελέσιμο.

Άσκηση 2: (0.5) Έστω S ένα σύνολο n ξένων (μη τεμνόμενων) ευθυγράμμων τμημάτων στο επίπεδο και p ένα σημείο που δεν περιέχεται σε κανένα από αυτά. Θέλουμε να προσδιορίσουμε όλα τα τμήματα του S που είναι ορατά από το p, δηλαδή όλα τα τμήματα του S που περιέχουν κάποιο σημείο q τέτοιο ώστε το ανοικτό τμήμα που ορίζεται από το p και q να μην τέμνει κανένα ευθύγραμμο τμήμα του S. Καταστρώστε έναν αλγόριθμο που να επιλύει αυτό το πρόβλημα σε χρόνο O(nlogn) χρησιμοποιώντας μία περιστρεφόμενη ημιευθεία με αρχή το p (σάρωση επιπέδου αλλά με περιστροφή).

Άσκηση 2.14 από το βιβλίο



<u>Άσκηση 3:</u> (0.4 μονάδες)

α) Να αποδείξετε το θεώρημα του κάστρου: Για την επιτήρηση της **εξωτερικής περιοχής** ενός πολυγώνου P με n κορυφές αρκεί να τοποθετήσουμε $\left\lceil \frac{n}{2} \right\rceil$ κάμερες οπουδήποτε στο σύνορό του, και μάλιστα υπάρχουν παραδείγματα (να δώσετε ένα) όπου τόσες κάμερες είναι απαραίτητες.

β) Έστω ότι μπορούμε να τοποθετήσουμε κάμερες σε οποιοδήποτε σημείο στο εξωτερικό του πολυγώνου P. Δώστε ένα παράδειγμα όπου $\left\lceil \frac{n}{3} \right\rceil$ κάμερες πρέπει να χρησιμοποιηθούν ώστε να επιτηρηθεί όλη η εξωτερική περιοχή του P.

<u>Άσκηση 4:</u> (0.3 μονάδες)

Δείξτε ότι αν ένα πολύγωνο έχει O(1) κορυφές στροφής τότε ο αλγόριθμος που παρατέθηκε στο κεφάλαιο 3 για τριγωνοποίηση πολυγώνου μπορεί να τροποποιηθεί ώστε να έχει χρόνο εκτέλεσης $O(n)$. (Άσκηση 3.9)

<u>Άσκηση 5:</u> (1 μονάδες)

Στην ιστοσελίδα του βιβλίου Computational Geometry in C ([1] στην ιστοσελίδα του μαθήματος από όπου μπορείτε να βρείτε και την ιστοσελίδα του βιβλίου) υπάρχει κώδικας σε Java ενός applet με υλοποιημένους κάποιους αλγόριθμους. Θα πρέπει να υλοποιήσετε τον τυχαιοκρατικό αλγόριθμο γραμμικού προγραμματισμού που προτείνεται στο βιβλίο (Κεφάλαιο 4.4) για την εύρεση τομών ημιεπιπέδων σε αυτό το γραφικό περιβάλλον. Θα πρέπει να «πειράξετε» το περιβάλλον ώστε να μπορείτε να σχεδιάσετε τέτοια ημιεπίπεδα. Το παραδοτέο θα είναι όλος ο κώδικας με το αντίστοιχο εκτελέσιμο. (αν κάνατε την αντίστοιχη άσκηση από το πρώτο σετ τότε να κάνετε την υλοποίηση σε αυτό το εκτεταμένο περιβάλλον).

<u>Άσκηση 6:</u> (0.5 μονάδες)

Σε n παράλληλες σιδηροδρομικές γραμμές κινούνται n τρένα με σταθερές ταχύτητες v_1, v_2, …, v_n. Τη χρονική στιγμή $t = 0$ τα τρένα βρίσκονται στις θέσεις k_1, k_2, …, k_n. Χρησιμοποιώντας τον αλγόριθμο για τον υπολογισμό της τομής ημιεπιπέδων, σχεδιάστε έναν αλγόριθμο χρόνου $O(n\log n)$ που να βρίσκει όλα τα τρένα που προπορεύονται σε κάποια χρονική στιγμή.

Άσκηση 7: (0.4 μονάδες)

Να αποδείξετε ότι η εύρεση της περιοχής τομής *n* ημιεπιπέδων στον διδιάστατο χώρο έχει πολυπλοκότητα $\Omega(n\log n)$.

Άσκηση 8: (0.4 μονάδες) Άσκηση 5.13 του βιβλίου.

Άσκηση 9: (0.4 μονάδες) Άσκηση 6.2 του βιβλίου.

Άσκηση 10: (0.3 μονάδες) Άσκηση 10.10 του βιβλίου.

Άσκηση 11: (0.3 μονάδες) Άσκηση 8.14 του βιβλίου.

Άσκηση 12: (1 μονάδα)

Στην ιστοσελίδα του βιβλίου Computational Geometry in C ([1] στην ιστοσελίδα του μαθήματος από όπου μπορείτε να βρείτε και την ιστοσελίδα του βιβλίου) υπάρχει κώδικας σε Java ενός applet με υλοποιημένους κάποιους αλγόριθμους. Θα πρέπει να υλοποιήσετε τον τυχαιοκρατικό αυξητικό αλγόριθμο που προτείνεται στο βιβλίο (Κεφάλαιο 6.2) για τον εντοπισμό σημείου με χρήση τραπεζιομερών χαρτών. Θα πρέπει να «πειράξετε» το περιβάλλον ώστε να μπορείτε να σχεδιάσετε επίπεδες υποδιαιρέσεις. Το παραδοτέο θα είναι όλος ο κώδικας με το αντίστοιχο εκτελέσιμο. (αν κάνατε την αντίστοιχη άσκηση από τα πρώτα δύο σετ τότε να κάνετε την υλοποίηση σε αυτό το εκτεταμένο περιβάλλον).