Υλοποίηση και Οπτικοποίηση του Αλγορίθμου Υπολογισμού του Τριγωνισμού Delaunay

Διπλωματική Εργασία

Δημήτριος Σαμουρέλης

Επιβλέπων: Λεωνίδας Παληός

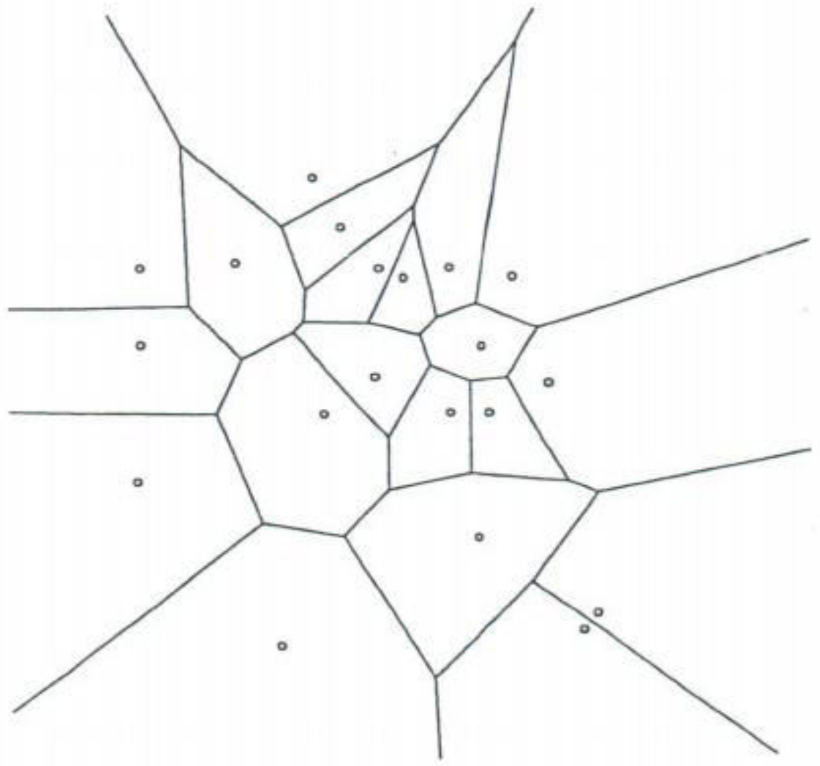
Ιωάννινα, Φεβρουάριος 2020

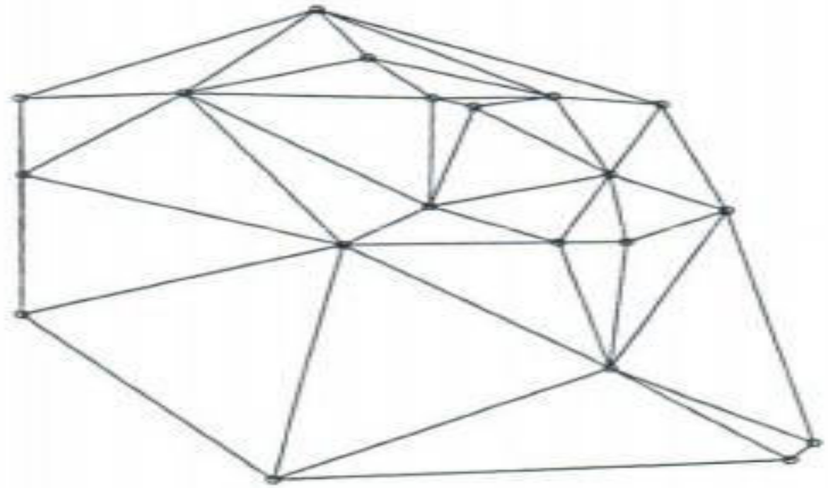
Τμήμα Μηχ. Η/Υ & Πληροφορικής
Πανεπιστήμιο Ιωαννίνων

Department of Computer Science & Engineering
University of Ioannina

Ο Τριγωνισμός Delaunay

- Έστω P ένα σύνολο n σημείων στο επίπεδο. Αυτά τα σημεία λέγονται κόμβοι. Σε κάθε ένα κόμβο αναθέτουμε όλα τα σημεία του επιπέδου που είναι πιο κοντά σε αυτόν παρά σε οποιοδήποτε άλλο κόμβο σύμφωνα με την Ευκλείδεια απόσταση. Τα σημεία αυτά σχηματίζουν την περιοχή Voronoi του κόμβου.
- Το διάγραμμα Voronoi και ο Delaunay τριγωνισμός περιέχουν την ίδια πληροφορία την οποία αναπαριστούν σε διαφορετική μορφή.





Ιδιότητες

- Ο T είναι ένας τριγωνισμός Delaunay του P αν και μόνο αν κανένα σημείο του P δεν κείται στο εσωτερικό του περιγεγραμμένου κύκλου κάποιου τριγώνου του T.
- Κάθε τριγωνισμός Delaunay του P μεγιστοποιεί την ελάχιστη γωνία ως προς όλους τους τριγωνισμούς του P.

Στόχος

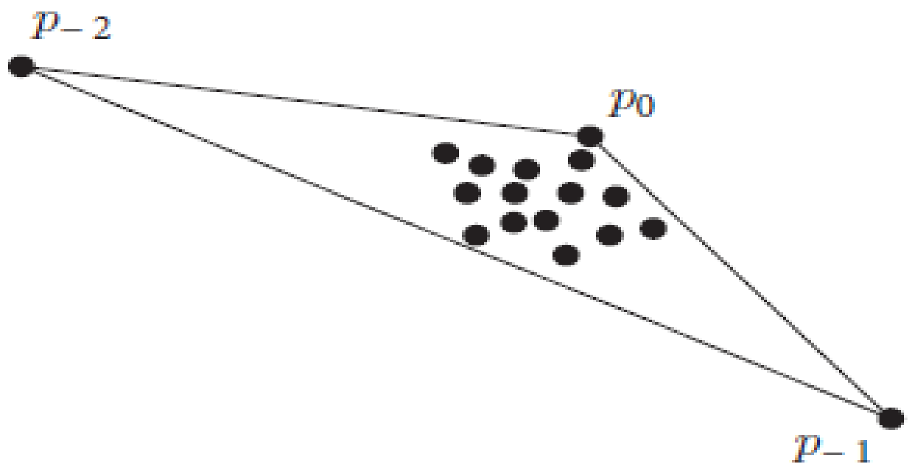
- Η μελέτη και η υλοποίηση του αλγορίθμου των Guibas, Knuth και Sharir.
- Ο αλγόριθμος είναι τυχαιοκρατικός και αυξητικός.
- Υπολογίζεται ένας τριγωνισμός Delaunay του τρέχοντος σημειοσυνόλου.

Εφαρμογές

- Σε εφαρμογές της μεθόδου πεπερασμένων στοιχείων.
- Στην αναπαράσταση 3Δ Χαρτών.
- Στον υπολογισμό του Ευκλείδειου ελάχιστου γενετικού δέντρου (EMST).

Ο Αλγόριθμος(i)

- Έστω p_0 το λεξικογραφικά υψηλότερο σημείο του P.
- Έστω p_{-1} και p_{-2} δύο σημεία στο $\mathbb{R}^2$ σε αρκετά μεγάλη απόσταση και τέτοια ώστε το P να εμπεριέχεται στο τρίγωνο $p_0p_{-1}p_{-2}$.
- Δημιουργούμε τον τριγωνισμό T που αποτελείται από το μοναδικό τρίγωνο $p_0p_{-1}p_{-2}$.
- Υπολογίζουμε μια τυχαία μετάθεση p_1, p_2,…,p_n του P \ {p_0}.
- για r ← 1 έως n
    - Εισάγουμε το p_r στον T
    - Βρίσκουμε ένα τρίγωνο $p_ip_jp_k$ ∈ T που περιέχει το p_r.



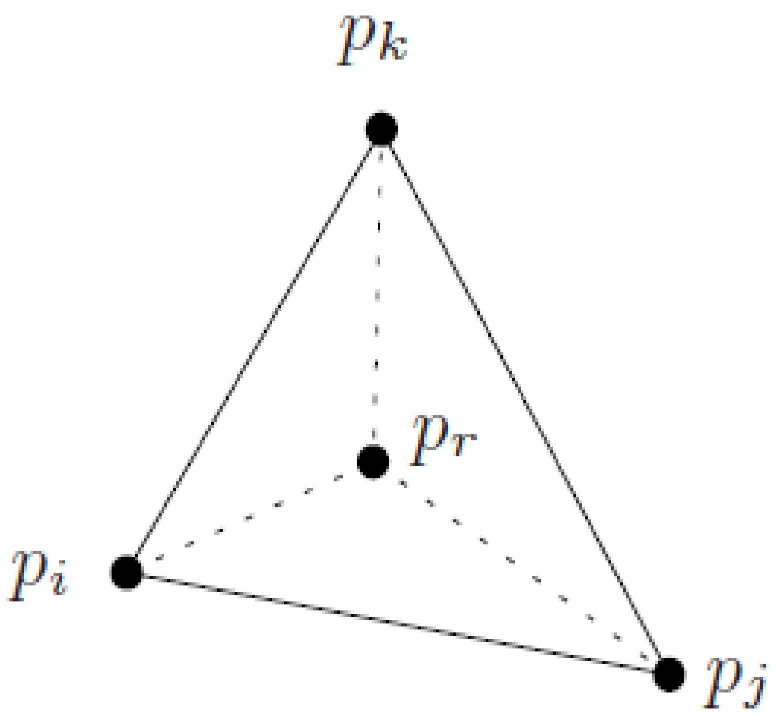
Ο Αλγόριθμος(ii)

εάν το p_r κείται στο εσωτερικό του τριγώνου $p_i p_j p_k$

τότε

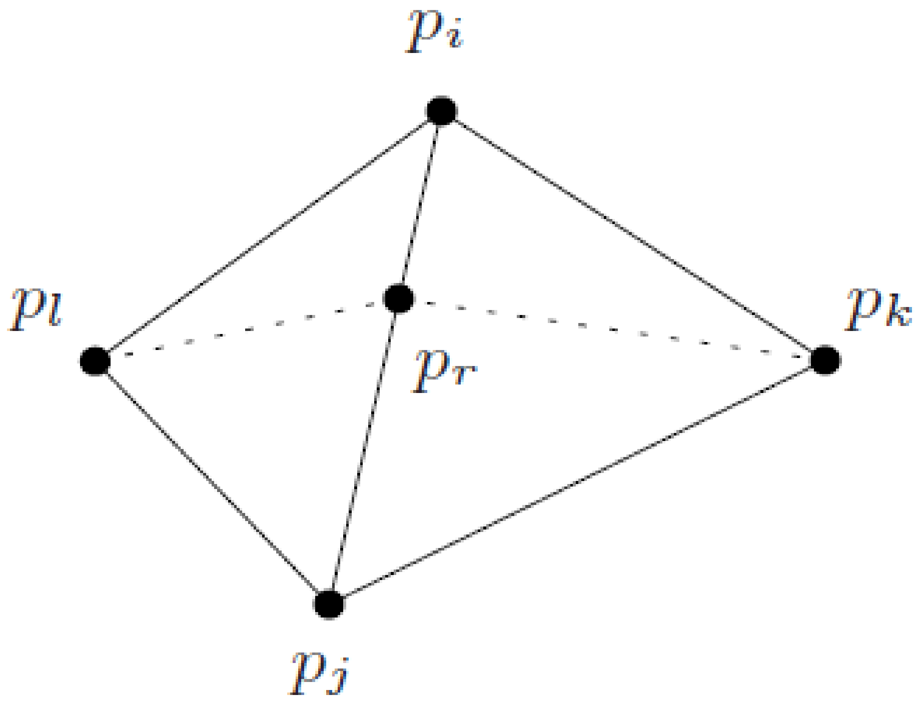
- Προσθέτουμε ακμές από το p_r προς τις τρεις κορυφές του $p_i p_j p_k$, διασπώντας έτσι το $p_i p_j p_k$ σε τρία τρίγωνα.
- ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, $p_i p_j$, T)
- ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, $p_j p_k$, T)
- ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, $p_k p_i$, T)

άλλως



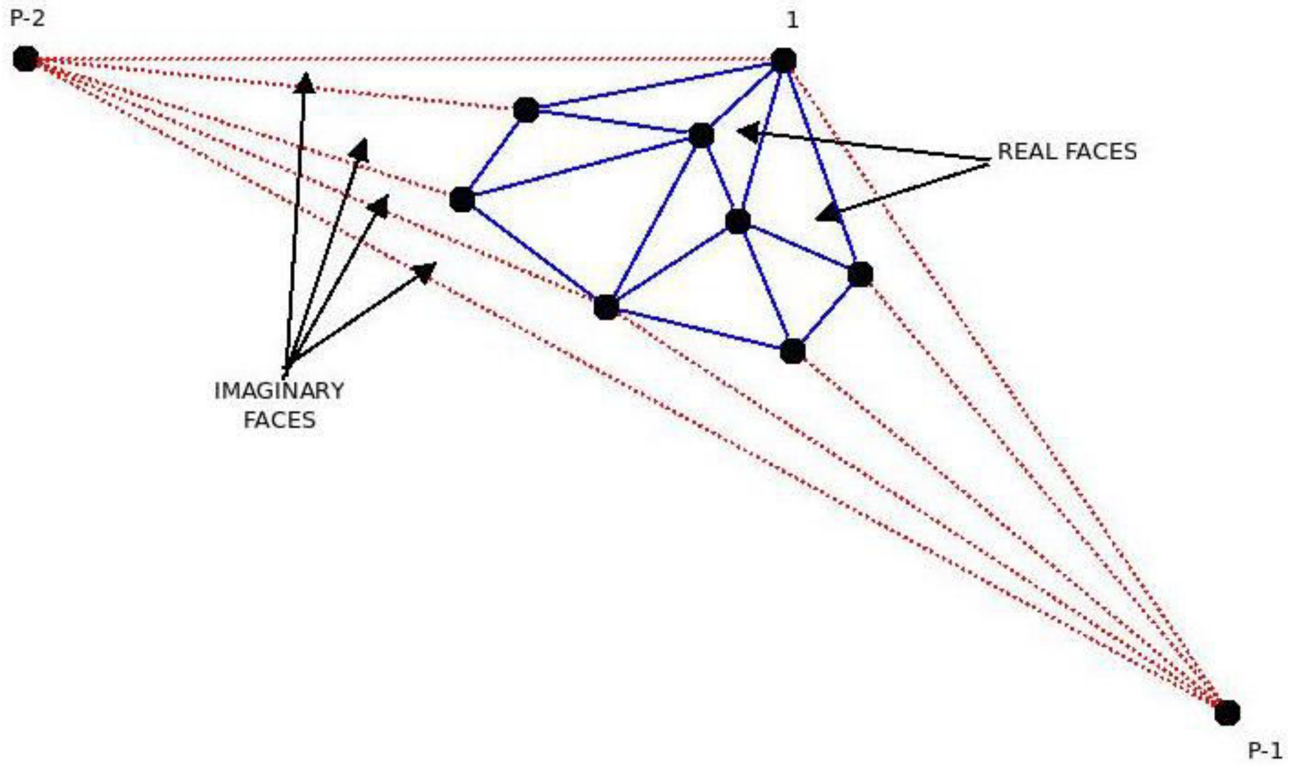
Ο Αλγόριθμος(iii)

- ($\ast$ το p_r κείται επάνω σε ακμή του $p_ip_jp_k$, έστω στην ακμή p_ip_j $\ast$)
    - Προσθέτουμε ακμές από το p_r προς το p_k και προς την τρίτη κορυφή p_l του άλλου τριγώνου που προσπίπτει στην p_ip_j, διασπώντας έτσι τα δύο τρίγωνα που προσπίπτουν στην p_ip_j σε τέσσερα τρίγωνα.
    - ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, p_ip_l, T)
    - ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, p_lp_j , T)
    - ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, p_jp_k, T)
    - ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, p_kp_i, T)



Ο Αλγόριθμος(iv)

- Διαγράφουμε από τον T τα σημεία p_{-1} και p_{-2}, και όλες τις προσπίπτουσες σε αυτά ακμές.
- Επιστροφή T



Η Συνάρτηση για την Διόρθωση Ακμής

ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ($P_{r,}$ $P_iP_{j,}$ T)

(∗ Το εισαγόμενο σημείο είναι το p_r, και η p_ip_j είναι η ακμή του T που πιθανόν να πρέπει να μεταστραφεί. ∗)

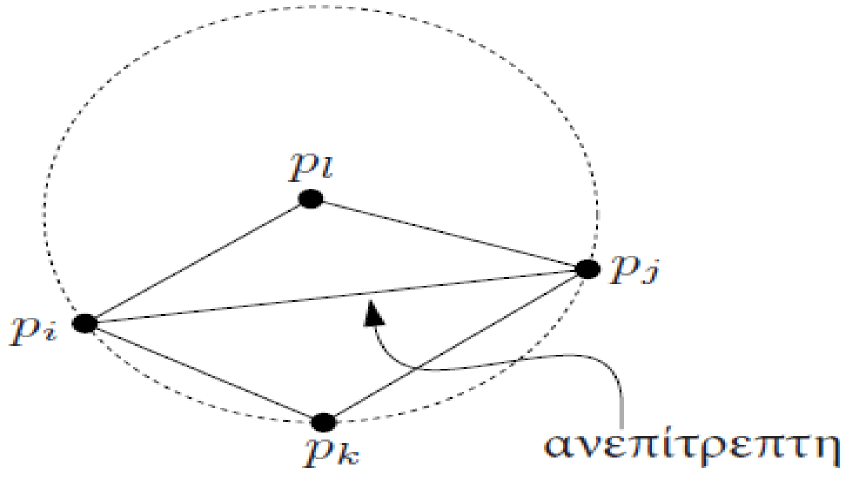
εάν η p_ip_j είναι ανεπίτρεπτη

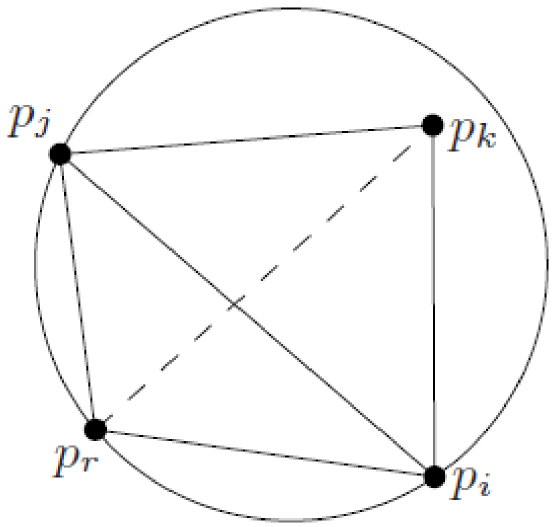
τότε

- Έστω $p_ip_jp_k$ το τρίγωνο που είναι παρακείμενο του $p_rp_ip_j$ με κοινή πλευρά την p_ip_j .

(∗ Μεταστροφή της p_ip_j: ∗)

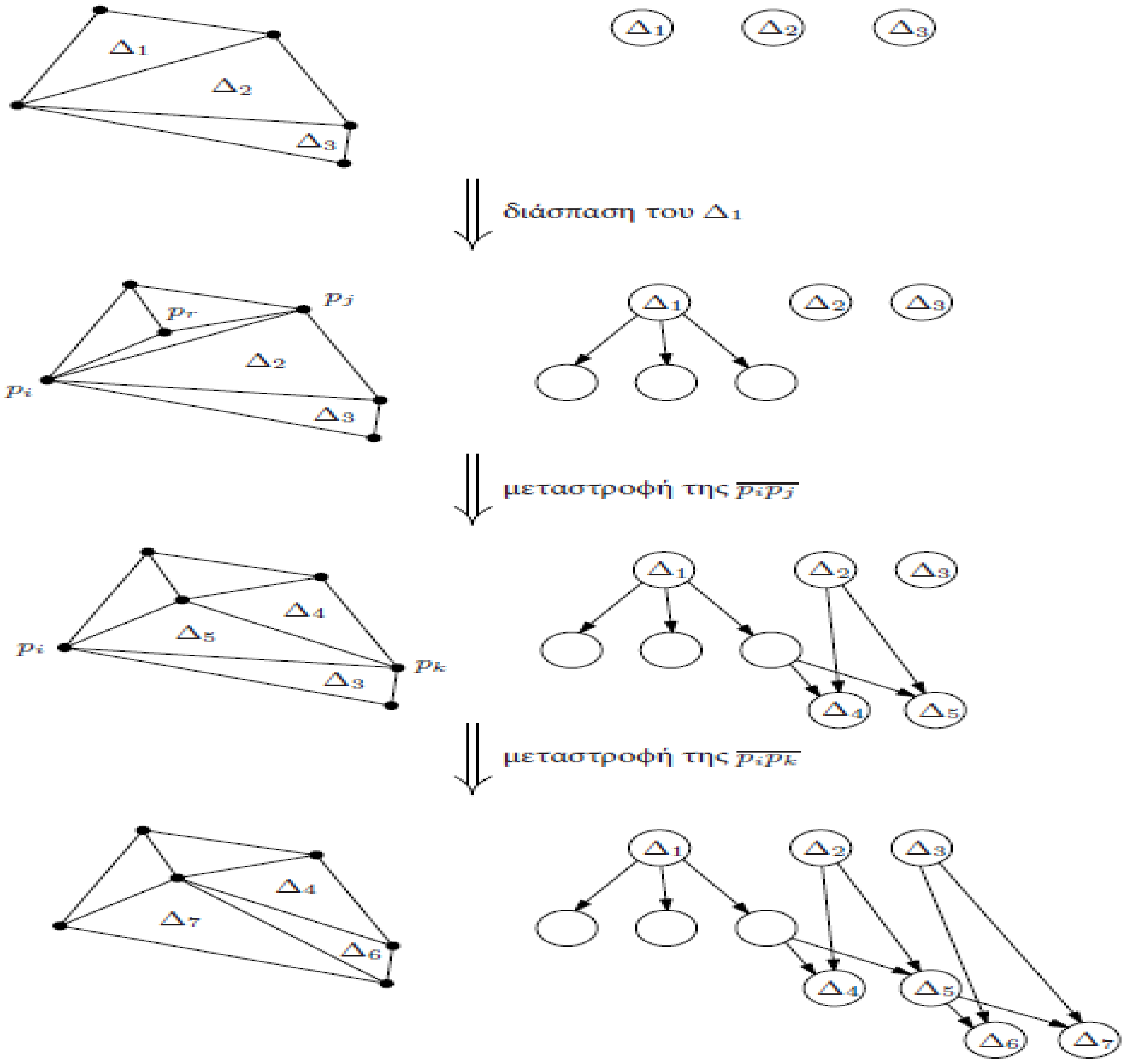
- Αντικαθιστούμε την p_ip_j με την p_rp_k.
- ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, p_ip_k, T)
- ΔΙΟΡΘΩΣΗ_ΑΚΜΗΣ (p_r, p_kp_j , T)



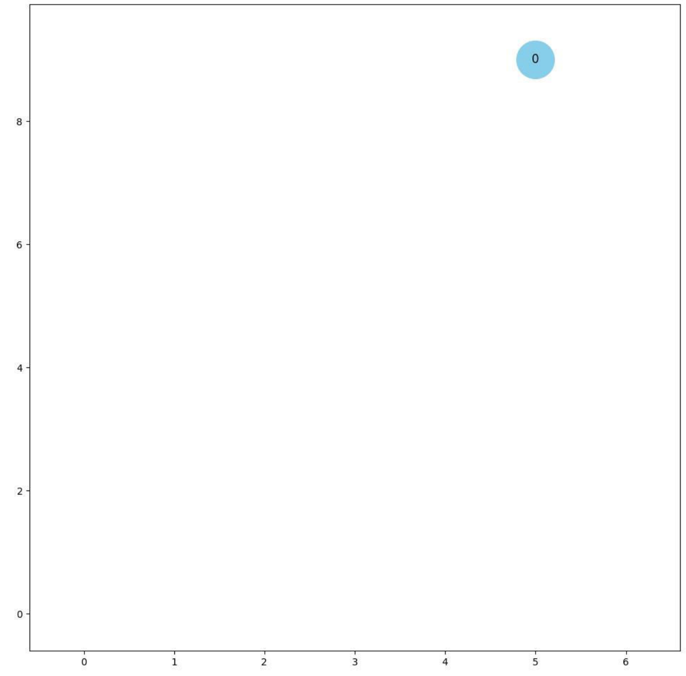


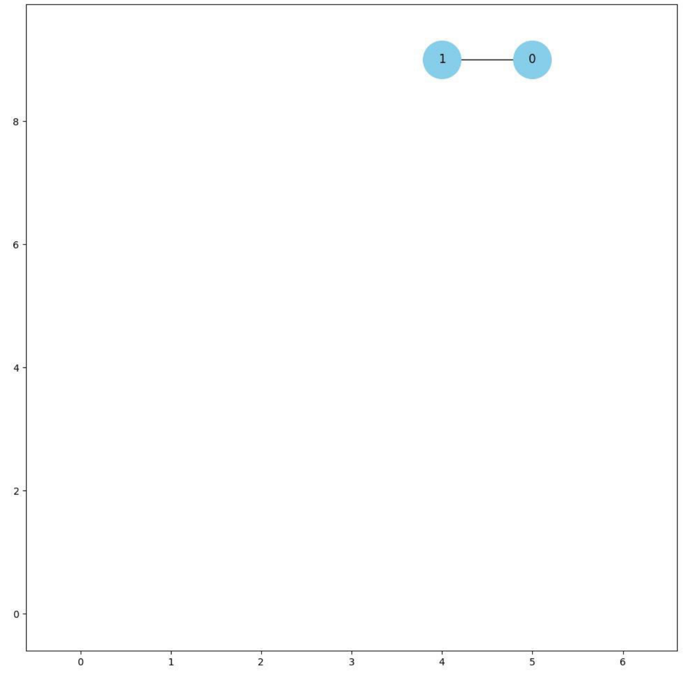
Μεταβολή Δέντρου Αναζήτησης

- Τα φύλλα της D αντιστοιχούν στα τρίγωνα του τρέχοντος τριγωνισμού Τ.
- Κατασκευάζουμε αμφίδρομους δείκτες μεταξύ των φύλλων και του τριγωνισμού.
- Οι εσωτερικοί κόμβοι της D αντιστοιχούν σε τρίγωνα που περιλαμβάνονταν στον τριγωνισμό σε κάποιο προηγούμενο στάδιο, αλλά έχουν ήδη καταστραφεί.

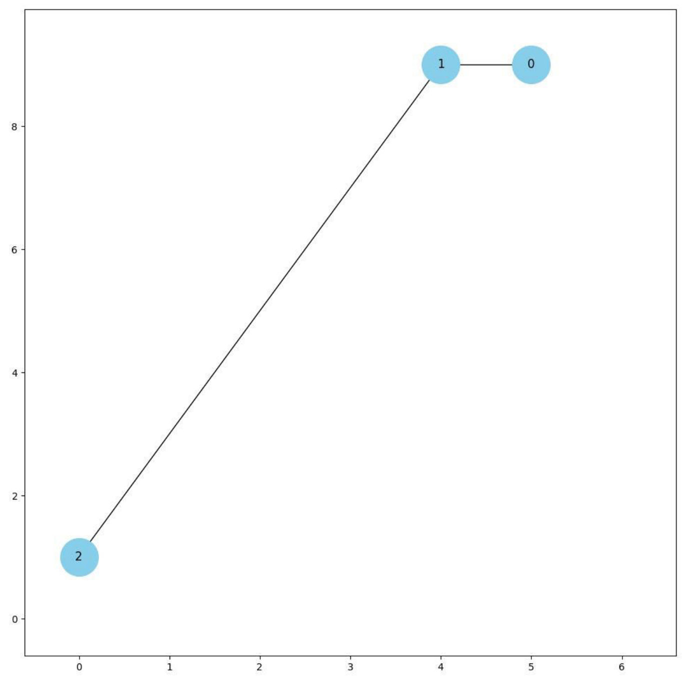


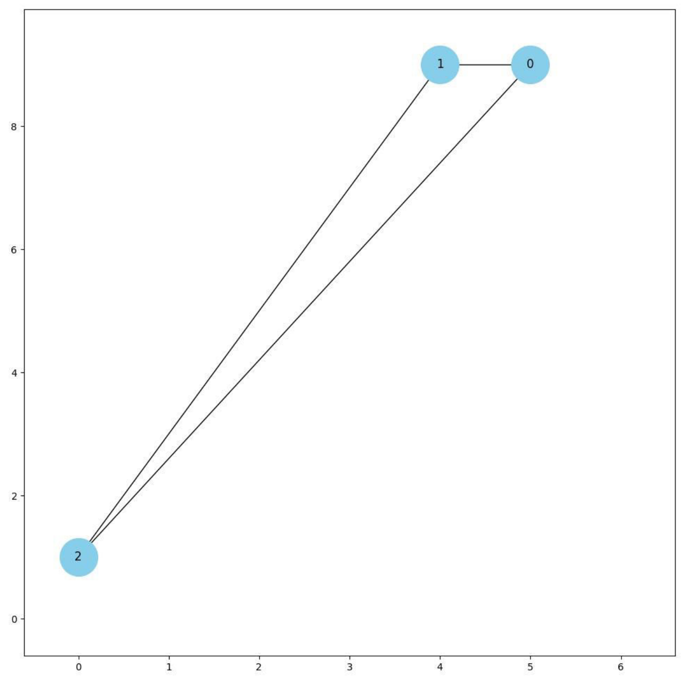
Παράδειγμα Εκτέλεσης(i)



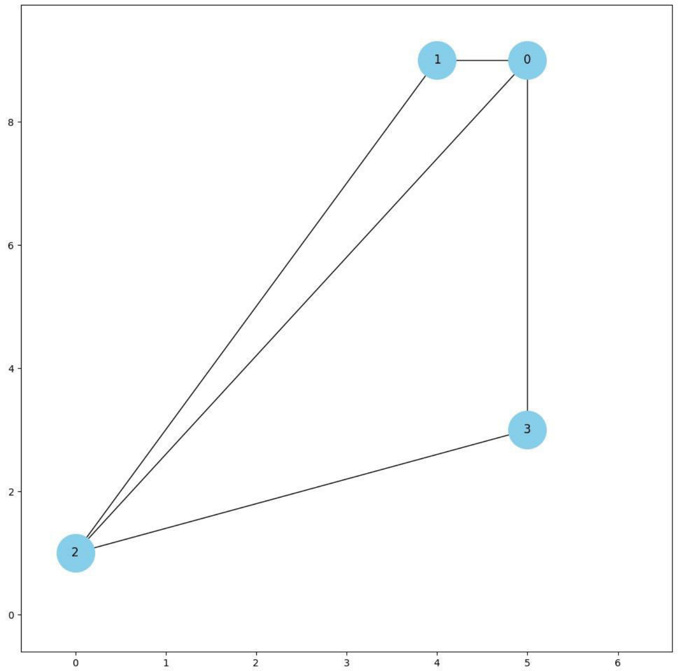


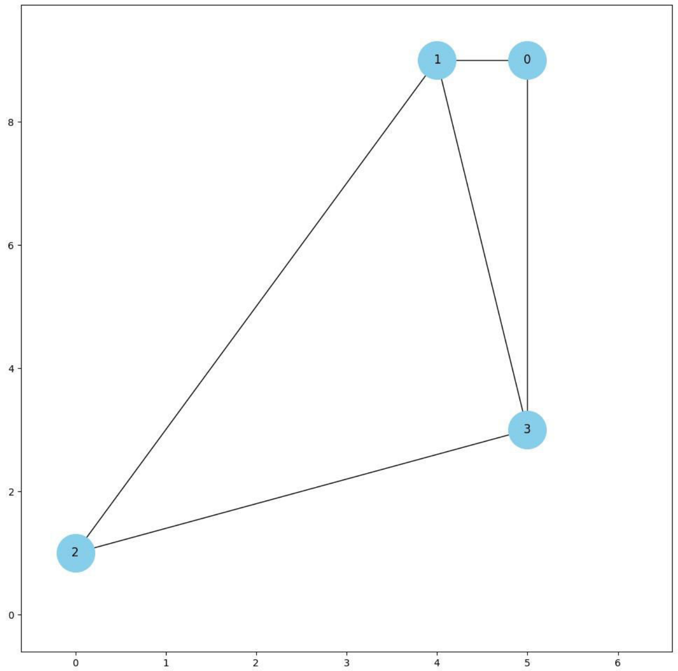
Παράδειγμα Εκτέλεσης(ii)



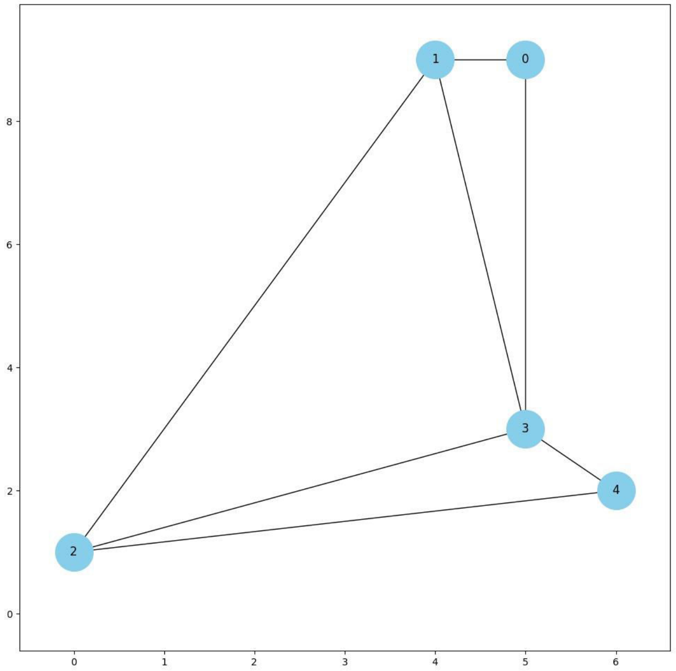


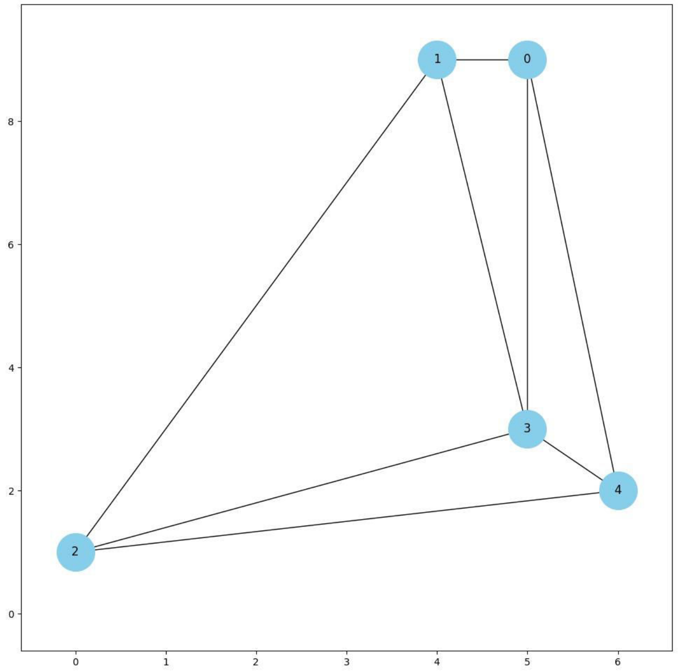
Παράδειγμα Εκτέλεσης(iii)



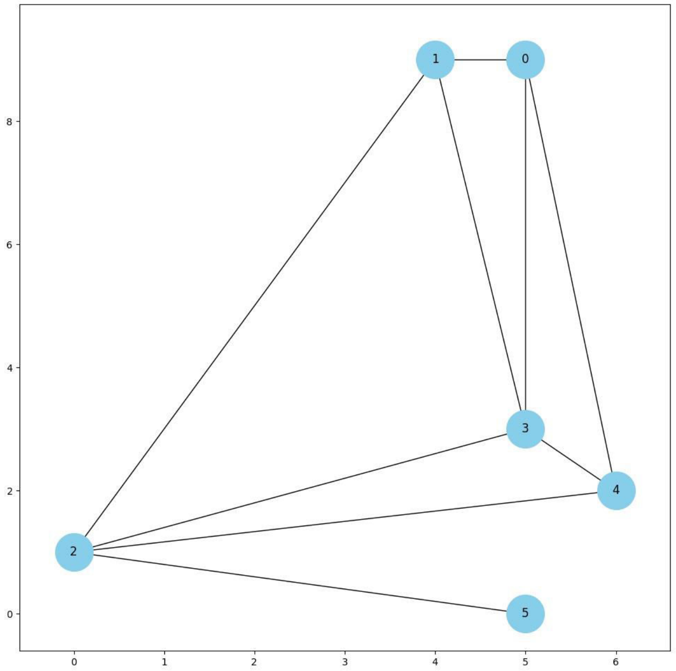


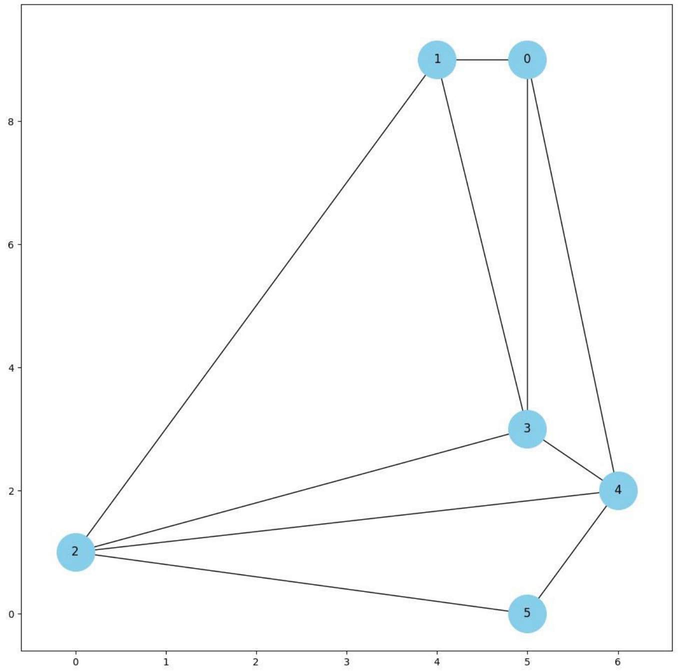
Παράδειγμα Εκτέλεσης(iv)



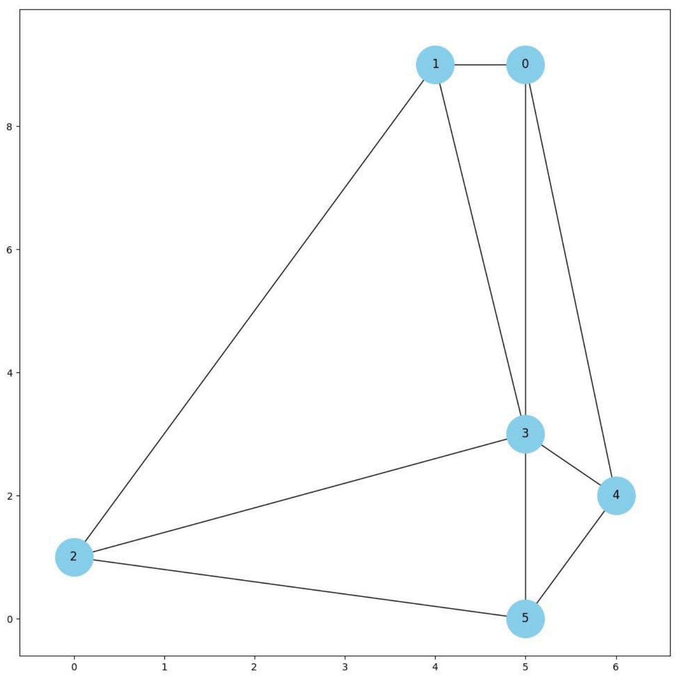


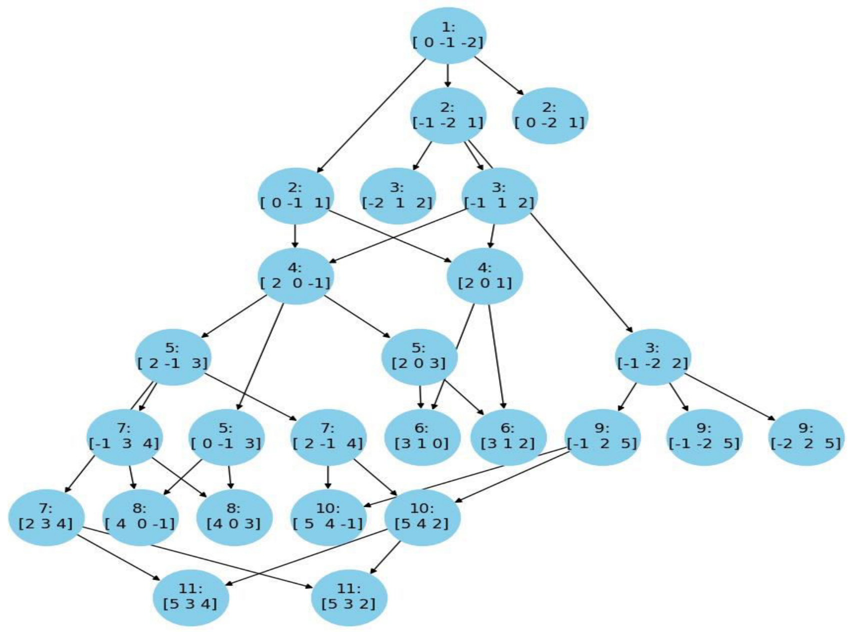
Παράδειγμα Εκτέλεσης(ν)





Παράδειγμα Εκτέλεσης(vi)





Πολυπλοκότητα

- Ο τριγωνισμός Delaunay ενός συνόλου P από n σημεία στο επίπεδο μπορεί να υπολογιστεί σε αναμενόμενο χρόνο O(nlogn), και σε αναμενόμενο χώρο O(n).
- Η αναζήτηση απαιτεί χρόνο γραμμικό ως προς το πλήθος των αποθηκευμένων στην δομή δέντρου τριγώνων που περιέχουν το p_{r}.

Κλάσεις Μέθοδοι(i)

Κλάση Point

- **__gt__** ώστε να ελεγχθούν δύο σημεία σύμφωνα με την λεξικογραφική τους διάταξη.

Κλάση Vector

- **innerProduct** για τον υπολογισμό του εσωτερικού γινομένου.
- **formAdjacentTrianglesForCheck** η οποία ελέγχει εάν η ακμή μαζί με αυτά τα δύο σημεία σχηματίζουν δύο παρακείμενα τρίγωνα.

Κλάση Triangle

- **contains** η οποία ελέγχει αν ένα σημείο βρίσκεται εντός του ορισμένου τριγώνου.

Κλάσεις Μέθοδοι(ii)

Κλάση Triangle

- **isInMinusNodeTriangle** για την περίπτωση που το τρίγωνο αποτελείται από κάποια ή και τις δύο αρνητικές κορυφές.
- **hasEdge** η οποία ελέγχει εάν μία ακμή είναι ακμή του συγκεκριμένου τριγώνου.
- **getThirdNode** επιστρέφει τον τρίτο κόμβο του τριγώνου με δεδομένο την ακμή που της δίνεται ως όρισμα

Κλάσεις Μέθοδοι(iii)

Κλάση Circle

- **contains** η οποία ελέγχει κατά πόσο ένα σημείο βρίσκεται εντός του κύκλου που κατασκευάστηκε.

Κλάση Dtree

- **checkChildrenForTriangle** δέχεται ως όρισμα ένα τρίγωνο και επιστρέφει τον κόμβο του δέντρου από τα παιδιά του που το περιέχει.
- **getNodeWithTriangle** δέχεται ως όρισμα ένα τρίγωνο και ελέγχει εάν ο κόμβος το περιέχει.
- **getLastContainingTriangle** δέχεται ως όρισμα ένα σημείο επιστρέφει το μικρότερο τρίγωνο του δέντρου που το περιέχει και την ακμή του τρίγωνου.

Κλάσεις Μέθοδοι(iv)

Κλάση Dcel

- **execute** με την οποία γίνεται εκτέλεση του αλγορίθμου Delaunay.
- **__get_initial_point** για την επιλογή του λεξικογραφικά ψηλότερου σημείου.
- **__fix_edge** για τη διόρθωση ακμής.
- **__shift_edge** για τη μεταστροφή ακμής.

Επεκτάσεις

- Να υλοποίηθει ένα κατάλληλο γραφικό περιβάλλον στο οποίο ο κέρσορας του ποντικιού να αναγνωρίζει τη θέση του στο χώρο.