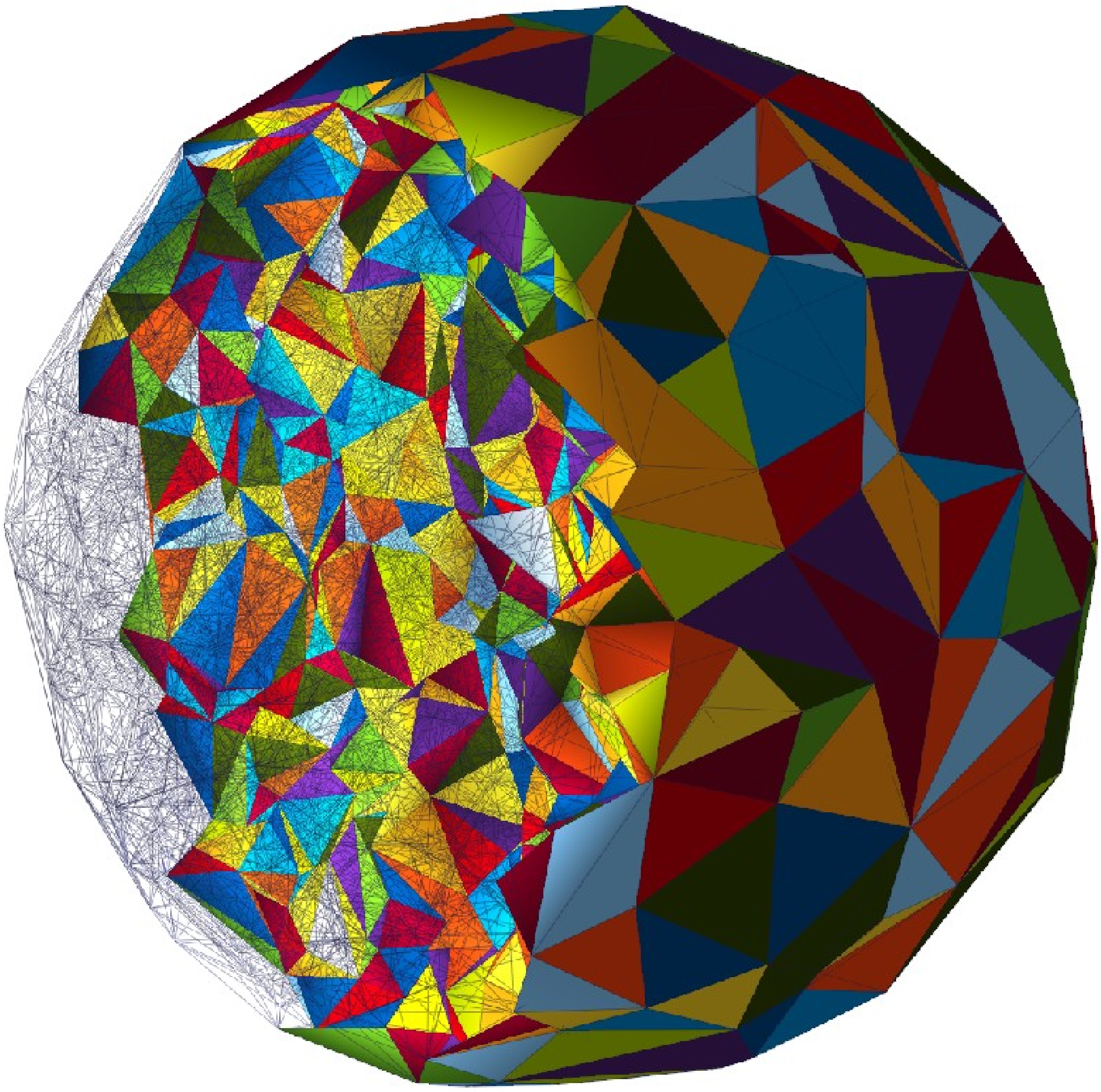
Υπολογιστική Γεωμετρία
Εργασία 2 (Delaunay)

ΒΛΑΣΣΗΣ ΠΑΝΑΓΙΩΤΗΣ
1115201400022

Εαρινό εξάμηνο
Ακαδ. Έτος: 2018-19



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

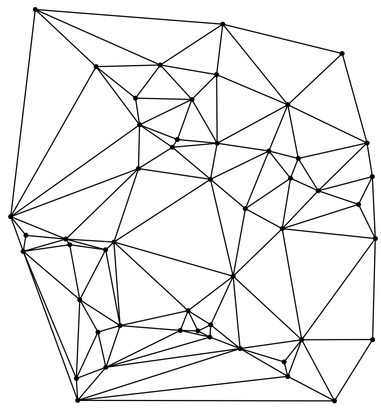
ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Περιεχόμενα

1. Γενικά – Τριγωνοποίηση

Ορισμός: <u>*Τριγωνοποίηση*</u> (triangulation) συνόλου σημείων ή εστιών στο επίπεδο καλείται ένα σύνολο τριγώνων με τις εξής ιδιότητες:

1. το σύνολο των κορυφών των τριγώνων ισούται με το σύνολο των δεδομένων εστιών,
2. η ένωση όλων των τριγώνων ισούται με το κυρτό περίβλημα των εστιών και
3. η τομή δύο τριγώνων αντιστοιχεί σε ακριβώς μία εκ των εξής περιπτώσεων: είναι κενή, ισούται με την κοινή κορυφή των δύο τριγώνων, ή ισούται με την κοινή τους ακμή.



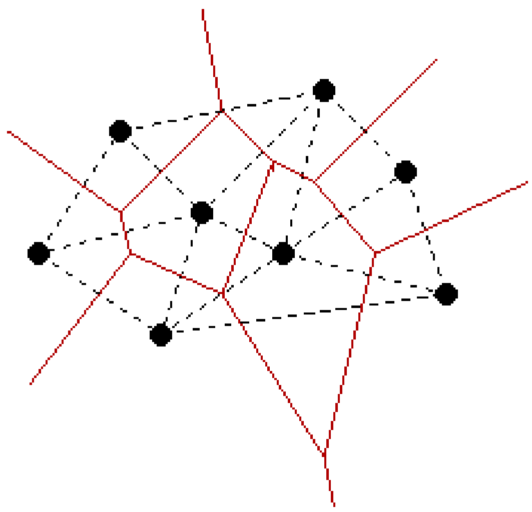
Εικόνα 1.1
Τριγωνοποίηση

Οι ιδιότητες 2 και 3 ορίζουν μια υποδιαίρεση (ή διαμέριση) του κυρτού τους περιβλήματος σε τρίγωνα. Η ιδιότητα 1 δηλώνει πως η τριγωνοποιήση αποτελεί ένα πολυγωνικό σύμπλοκο. Αφού τα τρίγωνα είναι άπλοκα, πρόκειται για πολυγωνικό απλοειδές σύμπλοκο. Η τριγωνοποίηση γενικεύεται αν αγνοήσουμε την πρώτη συνθήκη.

2. Τριγωνοποίηση Delaunay

Ορισμός: Η *τριγωνοποίηση Delaunay* δεδομένου συνόλου εστιών στο επίπεδο είναι ο δυϊκός γράφος του διαγράμματος Voronoi των εστιών:

1. Τα κελιά Voronoi αντιστοιχούν σε εστίες, δηλαδή κορυφές τριγώνων Delaunay.
2. Κάθε ζεύγος γειτονικών κελιών (δηλαδή ακμή Voronoi) αντιστοιχεί σε μία ακμή Delaunay, που ορίζεται από τις 2 εστίες: οι ευθείες των δύο ακμών είναι κάθετες μεταξύ τους.
3. Κάθε κορυφή Voronoi αντιστοιχεί σε ένα τρίγωνο Delaunay (ή κυρτό πολύγωνο)



Εικόνα 2.1
Δυϊκό διάγραμμα Voronoi - Delaunay

- *Θεώρημα*: Στην τριγωνοποίηση Delaunay ενός συνόλου εστιών:
  1. Τρεις εστίες ορίζουν τρίγωνο ανν ο περιγεγραμμένος ανοικτός κύκλος του είναι κενός, δηλαδή δεν περιέχει εστίες στο εσωτερικό του.
  2. Δύο εστίες ορίζουν ακμή ανν υπάρχει κενός κύκλος με τις εστίες στην περιφέρειά του, δηλαδή υπάρχει κλειστός δίσκος με χορδή η οποία ορίζεται από τις δύο εστίες και δεν περιέχει καμία άλλη εστία.

- *Λήμμα 1*: Η τριγωνοποίηση Delaunay είναι επίπεδος γράφος, με κόμβους τις κορυφές των τριγώνων → Γενικότερα, οποιαδήποτε τριγωνοποίηση είναι επίπεδος γράφος.
- *Λήμμα 2*: Έστω n μη συνευθειακές εστίες στο επίπεδο και k το πλήθος κορυφών στο κυρτό πολύγωνο τους. Τότε υπάρχουν 2n-k-2 τρίγωνα και 3n-k-3 ακμές σε κάθε τριγωνοποίηση των εστιών.
- *Θεώρημα*: Μία τριγωνοποίηση είναι έγκυρη ανν είναι Delaunay.
- *Πόρισμα*: Η τριγωνοποίηση Delaunay αντιστοιχεί στο λεξικογραφικά βέλτιστο διάνυσμα γωνιών.

Η τριγωνοποίηση Delaunay δεν είναι μοναδική αν ο δυϊκός γράφος Voronoi περιέχει κυρτά πολύγωνα τα οποία τριγωνοποιήθηκαν αυθαίρετα. Είναι δυνατό να αποδειχτεί πως όλες οι τριγωνοποιήσεις Delaunay οδηγούν στο ίδιο (βέλτιστο) διάνυσμα γωνιών.

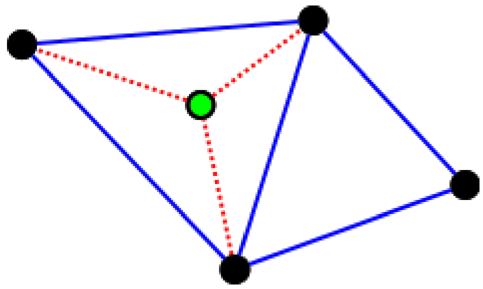
3. Αλγόριθμοι κατασκευής τριγωνοποίσης Delaunay

3.1 Αυξητικός αλγόριθμος (Incremental Algorithm) [1]

3.1.1 Περιγραφή αλγορίθμου

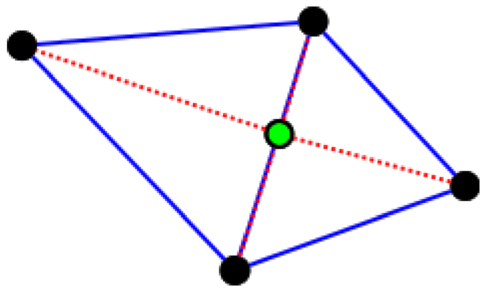
Ο αυξητικός αλγόριθμος υπολογίζει την τριγωνοποίηση δημιουργώντας ένα αρχικό τρίγωνο με 3 σημεία της τριγωνοποίησης και μετά προσθέτει τα υπόλοιπα σημεία. Για κάθε νέο σημείο δημιουργούνται νέα τρίγωνα, για τα οποία ελέγχουμε αν ικανοποιούνται οι συνθήκες της Delaunay τριγωνοποίησης. Αν ναι, συνεχίζουμε με το επόμενο σημείο, αλλιώς είναι απαιραίτητο να γίνουν flip μερικές ακμές της τριγωνοποίησης, προτού συνεχίσουμε με το επόμενο σημείο.

Για κάθε νέο σημείο, ο αλγόριθμος βρίσκει το τρίγωνο που περιέχει αυτό το σημείο, διαχωρίζει το τρίγωνο σε 3 (ή 4) νέα τρίγωνα και ελέγχει αν κάποιες από τις ακμές πρέπει να γίνουν flipped. Η διαδικασία διαχωρισμού του τριγώνου γίνεται ως εξής: Αν το νέο σημείο βρίσκεται εντός υπάρχοντος τριγώνου, τότε το τρίγωνο διαχωρίζεται σε 3 τρίγωνα. ***(Εικόνα 3.1.1)***



Εικόνα 3.1.1

Υπάρχει μία ειδική περίπτωση: αν το σημείο είναι συγραμμικό με κάποια ακμή τότε τα 2 τρίγωνα διαχωρίζονται σε 2 νέα, δημιουργώντας έτσι 3 νέα τρίγωνα. Αυτός ο διαχωρισμός φαίνεται οπτικά παρακάτω: ***(Εικόνα 3.1.2)***



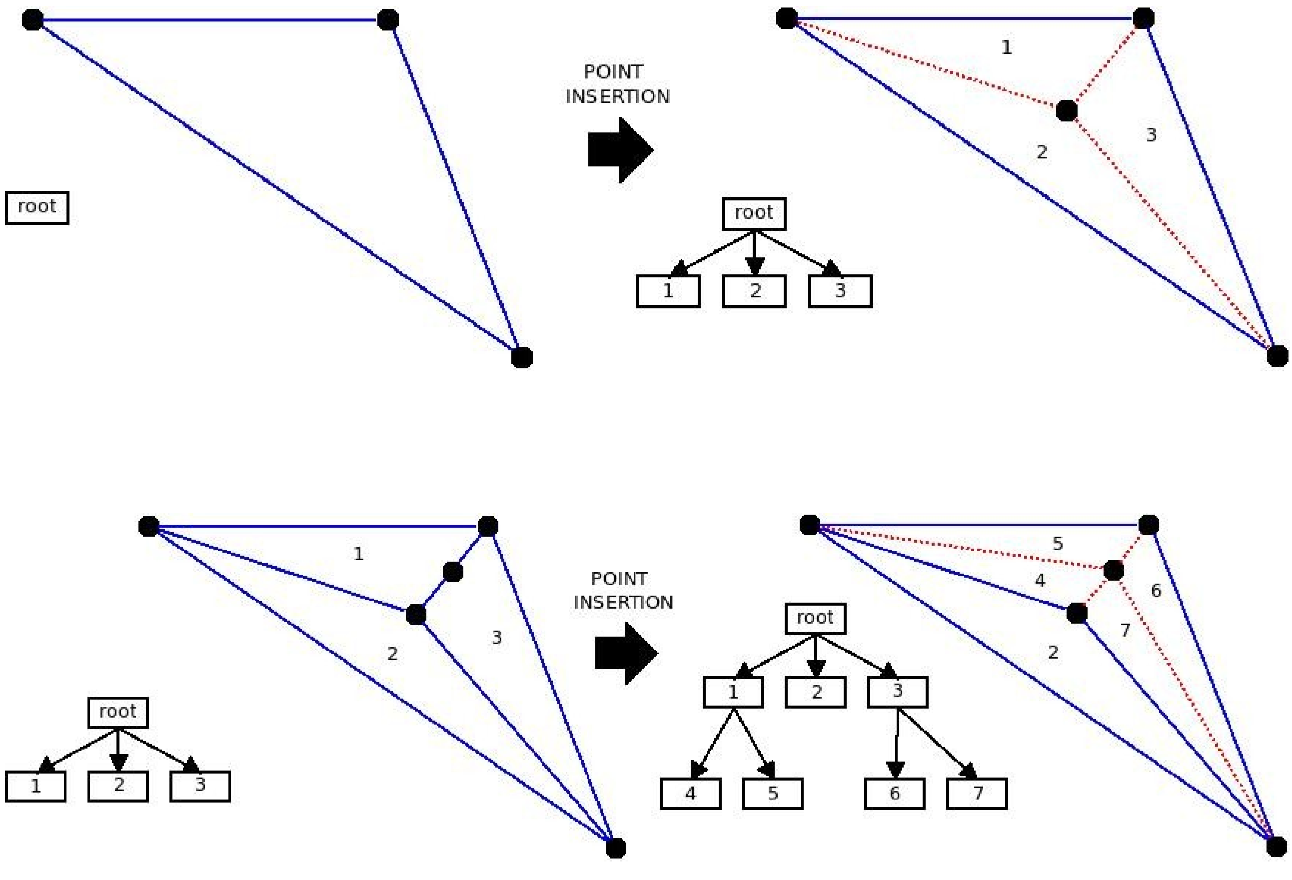
Εικόνα 3.1.2

3.1.2 Τρόπος υλοποίησης αλγορίθμου

Τα ζητήματα για τη απόδοση του αλγορίθμου είναι: ***(1)*** *<u>η τοποθεσία του τριγώνου που περιέχει το νέο σημείο</u> και* ***(2)*** *<u>ο αριθμός των ακμώς που πρέπει να γίνουν flipped αφού δημιουργηθούν τα νέα τρίγωνα</u>.*

Για την επίλυση του 1ου ζητήματος, δηλαδή να βρούμε το τρίγωνο που περιέχει το νέο σημείο, χρησιμοποιείται μία δομή δεδομένων τύπου Δέντρου (Tree). Όταν ένα νέο σημείο εισάγεται, βρίσκουμε το τρίγωνο στο οποίο περιέχεται και ο κόμβος στο δέντρο που έχει το τρίγωνο ανανεώνεται με την προσθήκη νέων κόμβων ως παιδιά του, κόμβοι που περιέχουν τα νέα τρίγωνα.

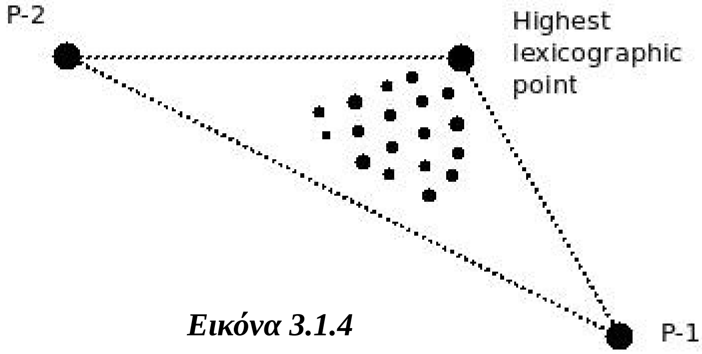
Οπότε, για τον αλγόριθμο έχουμε 2 δομές δεδομένων: DCEL (Doubly Connected Edge List),που αποθηκεύεται η τριγωνοποίηση και Δέντρο (Tree) που αποθηκεύονται όλα τα τρίγωνα που δημιουργούνται σε όλη τη φάση δημιουργίας της τριγωνοποίησης Delaunay.



Εικόνα 3.1.3

3.1.3 Βήματα υλοποίησης αλγορίθμου

Πρώτο βήμα *του αλγορίθμου είναι να δημιουργήσουμε ένα φανταστικό τρίγωνο, ισάγοντας το ως ρίζα στο δέντρο καθώς και στο DCEL. Αυτό το φανταστικό τρίγωνο φαίνεται στο παρακάτω σχήμα:* ***(Εικόνα 3.1.4)***

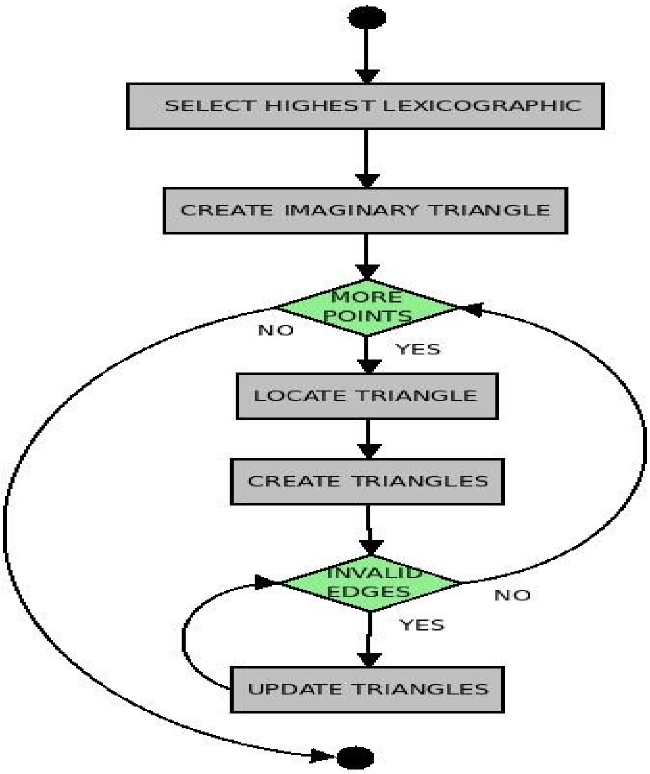


Εικόνα 3.1.4

Δεύτερο βήμα *είναι η εισαγωγή όλων των υπόλοιπως σημείων του συνόλου. Αυτό το βήμα υλοποιείται σε μερικά στάδια:*

- *Εντοπισμός τριγώνου που περιέχει το σημείο.*
- *Δημιουργία 3 (ή 4) νέων τριγώνων.*
- *Ανανέωση DCEL και Tree με τα νέα τρίγωνα.*
- *Έλεγχος αν μία ακμή δεν είναι Delaunay ακμή και, αν όχι, να γίνει flip.*

Το τελευταίο στάδιο μπορεί να μεταβάλλει την υπάρχουσα τριγωνοποίηση, με αποτέλεσμα να απαιτείται ο έλεγχος παραπάνω ακμών.

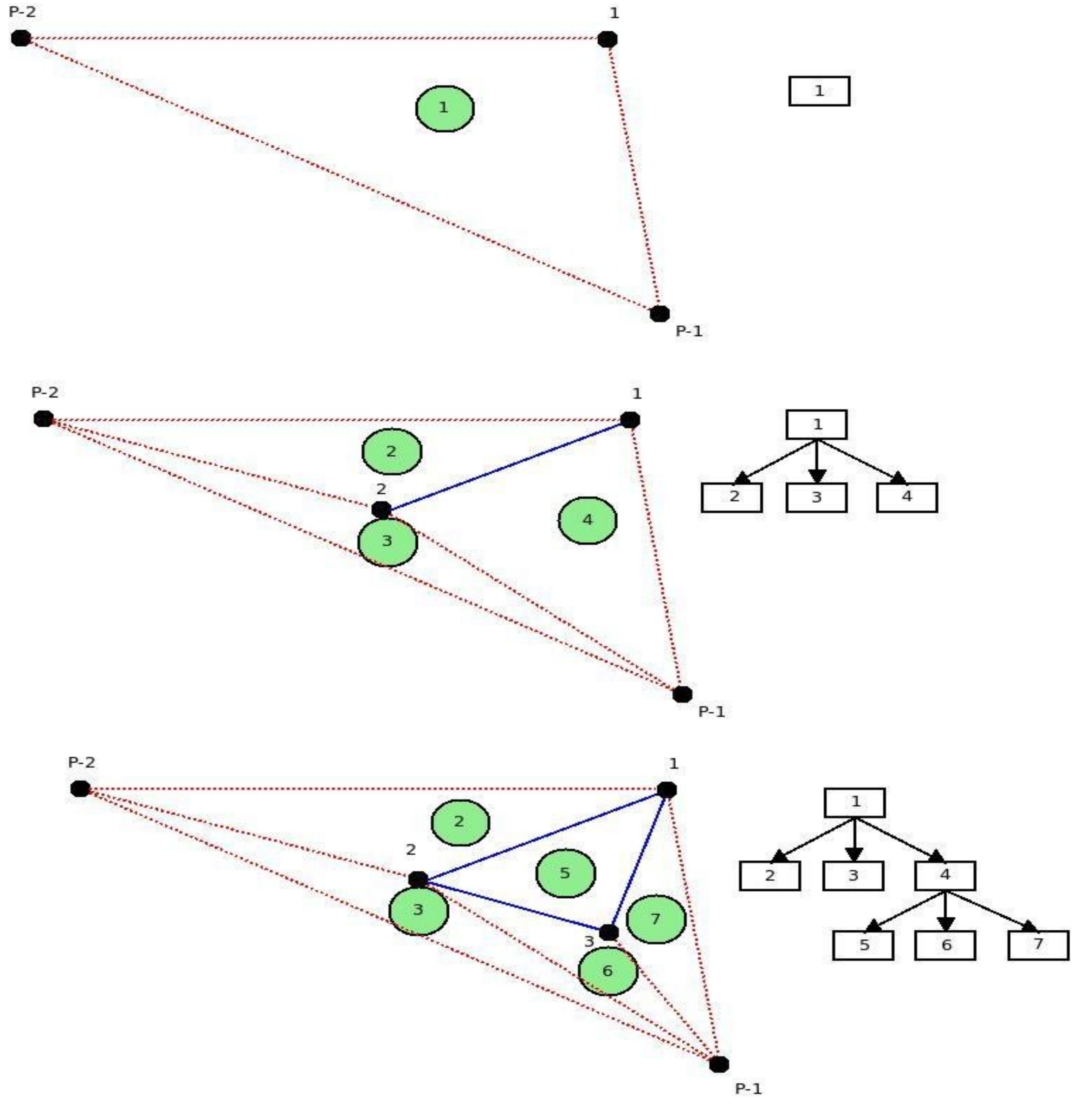


Πολυπλοκότητα: O(nlogn)

Εικόνα 3.1.5
Διάγραμμα ροής αυξητικού αλγορίθμου

3.1.4 Παράδειγμα εκτέλεσης αλγορίθμου

Οι παρακάτων εικόνες αναπαραστούν τις 3 πρώτες εισαγωγές στη τριγωνοποίηση. Όπως φαίνεται στο σχήμα, πριν υπάρξουν 3 πραγματικά σημεία στη τριγωνοποίηση υπάρχουν διάφορα τρίγωνα, από τα οποία μόνο ένα είναι πραγματικό. Η μόνη πραγματική όψη (face) είναι εκείνη με τον αριθμό 5, με τις υπόλοιπες να είναι φανταστικές (imaginary).



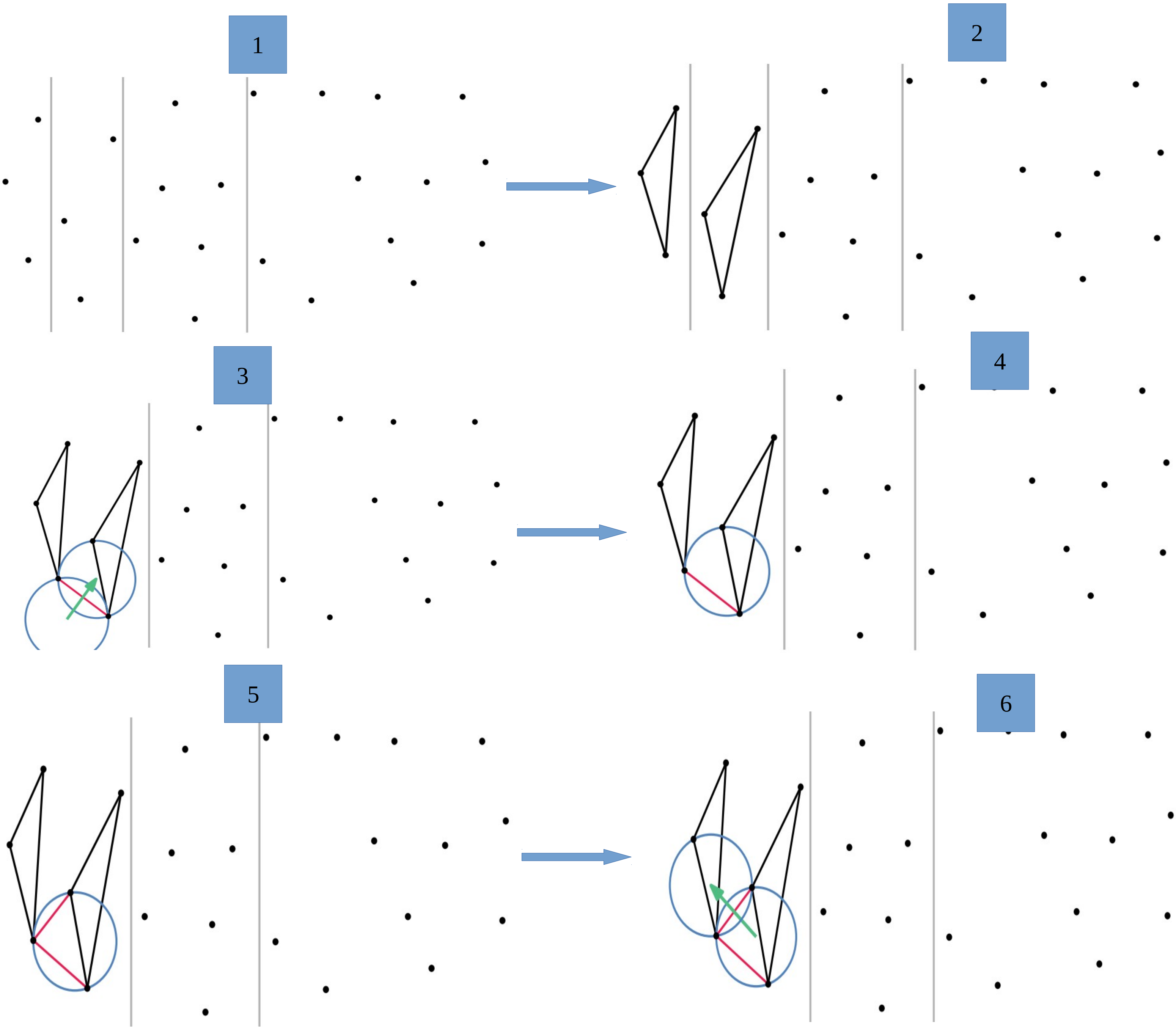
Εικόνα 3.1.6
Παράδειγμα εκτέλεσης αυξητικού αλγορίθμου

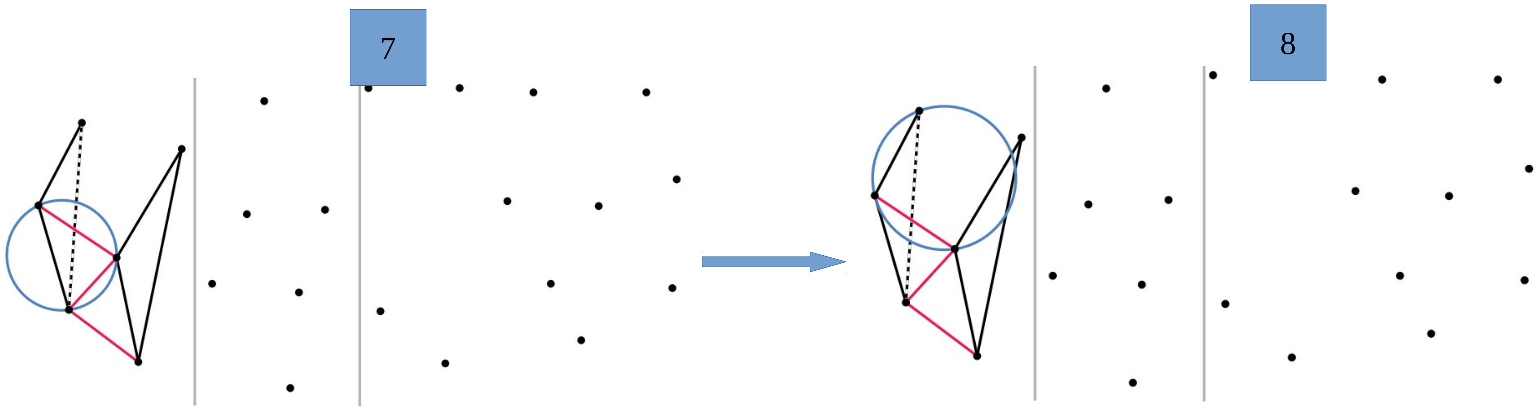
3.2 Αλγόριθμος Διαίρει και Βασίλευε (Divide and Conquer Algorithm) [2]

Για την εκτέλεση του αλγορίθμου Διαίρει και Βασίλευε απαιτούνται 2 βασικά βήματα κατά τη φάση της συγχώνευσης (merging):

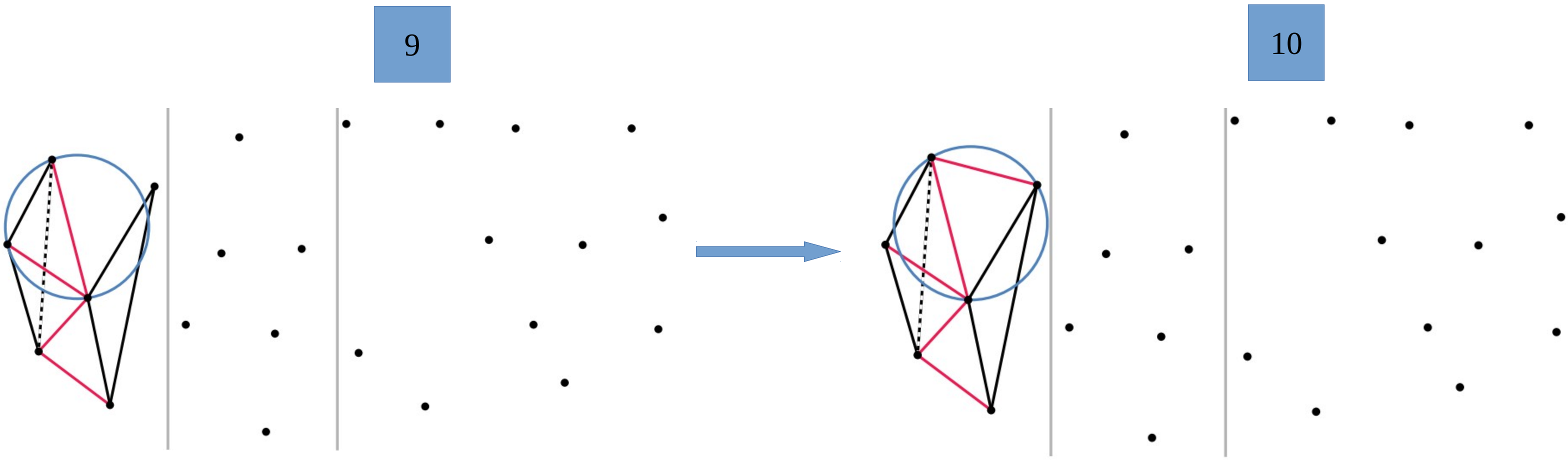
- Δημιουργία τριγώνων με κενούς περιγεγραμμένους κύκλους
- Διαγραφή συγκρουόμεων (conflicting) ακμών

Πολυπλοκότητα: O(nlogn)

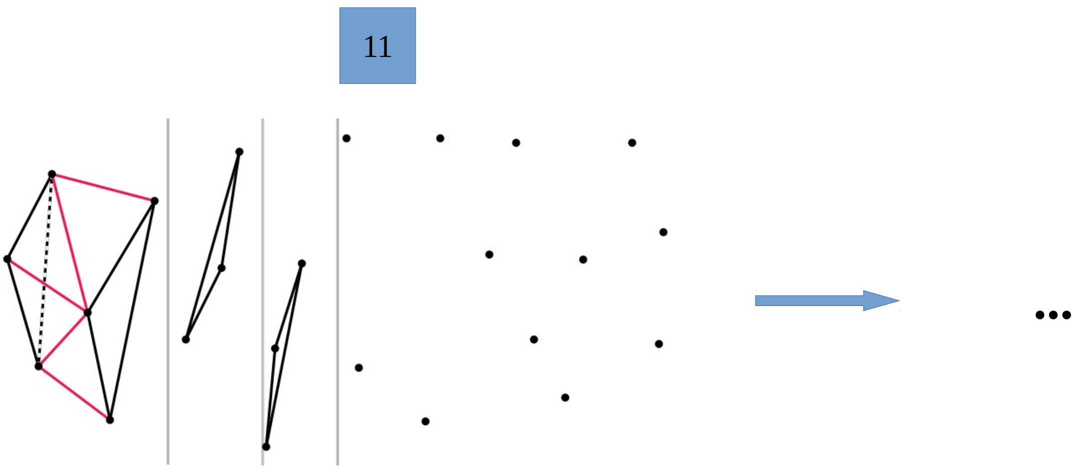




7
8

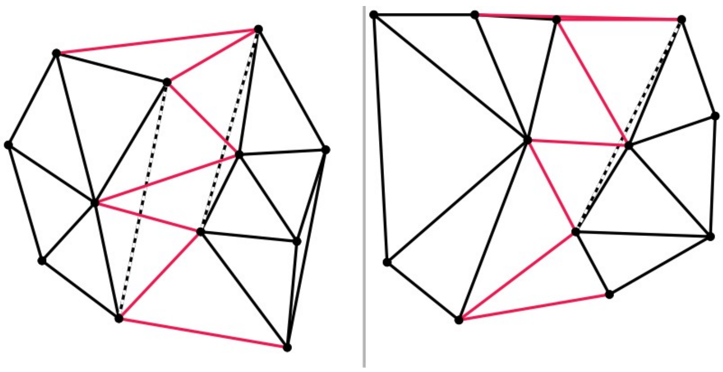


9
10

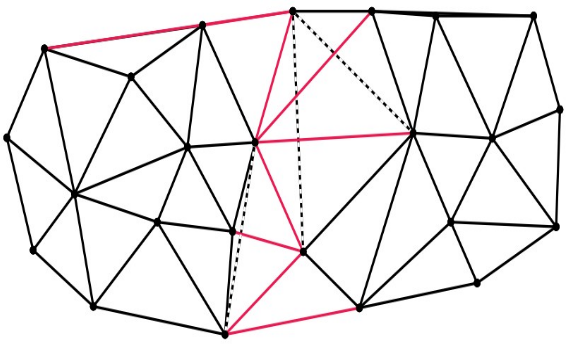


11
...

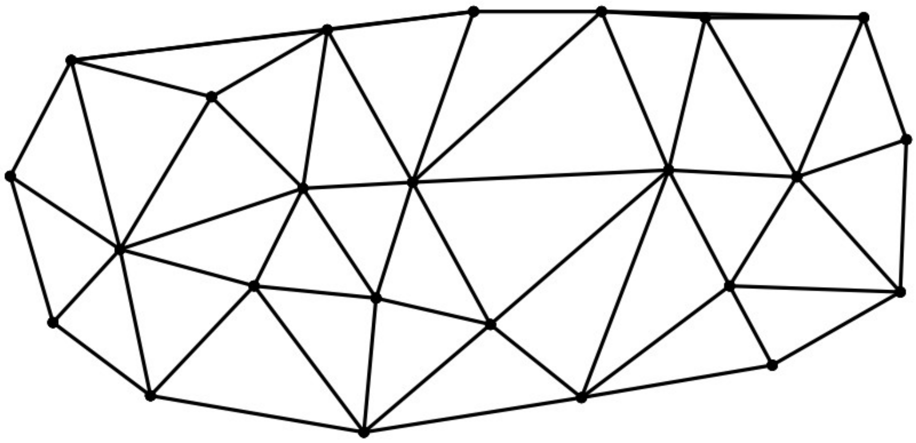
12



13



15



Τριγωνοποίηση Delaunay με Divide & Conquer

Βιβλιογραφία

[1] https://github.com/juannavascalvente/Delaunay/wiki/Incremental-Delaunay-triangulation

[2] http://www.ist.tugraz.at/_attach/Publish/Dcg/Delaunay_4.pdf

[3] https://www.comp.nus.edu.sg/~tants/gdel3d_files/3ddt.png

[4] *Γιάννης Ζ. Εμίρης, Υπολογιστική Γεωμετρία, σελ.200-211*