ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Μπεγέτης Νικόλαος

1115200700281

ΥΠΟΛΟΓΙΣΤΙΚΗ ΓΕΩΜΕΤΡΙΑ

Άσκηση 4(2η Προγραμματιστική): Τριγωνοποιήσεις στη CGAL

Όλα τα αρχεία και η αντιστοίχιση με τις ασκήσεις είναι τα εξής:

Άσκηση 1 → triangulation1.cpp

Άσκηση 2 → triangulation2.cpp

Άσκηση 3 → a) triangulation3.cpp

b) triangulation3_Reg_vs_Del.cpp

Bonus → triangulation_bonus.cpp

Για απορίες – ερωτήσεις πάνω στον κώδικά μου μπορείτε να μου στείλετε email στο sdi0700281@di.uoa.gr

Νικόλαος Μπεγέτης

Άσκηση 1:

Η άσκηση 1 έχει υλοποιηθεί ακριβώς όπως περιγράφεται από την εκφώνηση.

Όσον αφορά το ερώτημα: «Η τριγωνοποίηση που υπολογίζεται για το παραπάνω παράδειγμα είναι ισοδύναμη με την placing τριγωνοποίηση όπου η διάταξη (ordering) για το placing δίνεται από τους δείκτες των σημείων. Αυτό δεν ισχύει γενικά. Δείξτε το με ένα αντιπαράδειγμα.» η απάντηση σε αυτό το ερώτημα είναι ότι αφού η τριγωνοποίηση βασίζεται στην διάταξη των σημείων και άρα από την σειρά με την οποία διαβάζονται αν αλλάξουμε αυτή τη σειρά θα μπορούμε να διαπιστώσουμε αν προκύπτει η ίδια τριγωνοποίηση για όλες τις περιπτώσεις.

Όμως όπως έχουμε δείξει και στο μάθημα κάτι τέτοιο δεν ισχύει και μάλιστα από κάθε πολύπλευρο μπορούν να προκύψουν περισσότερες από μία τριγωνοποιήσεις. Για παράδειγμα για ένα τετράπλευρο όπως και αυτό που μας δίνεται στην εκφώνηση μπορούμε να έχουμε 2 τριγωνοποιήσεις, για κάθε

πεντάπλευρο μπορούμε να έχουμε 3 διαφορετικές τριγωνοποιήσεις κ.ο.κ. Συνεπώς, στην προκείμενη περίπτωση της άσκησης μας, αλλάζοντας τα σημεία μέσα στο αρχείο και τοποθετώντας τα με την εξής σειρά:

0.0 0.0

10.0 0.0

0.0 10.0

10.0 10.0

μπορούμε να πάρουμε την τριγωνοποίηση: {{1,2,3} {0,2,3}} που είναι η τριγωνοποίηση στο ίδιο τετράπλευρο αλλά με την αντίθετη διαγώνιο να το τριγωνοποιεί, αντί για αυτή που μας δόθηκε στην εκφώνηση.

Άσκηση 3(b):

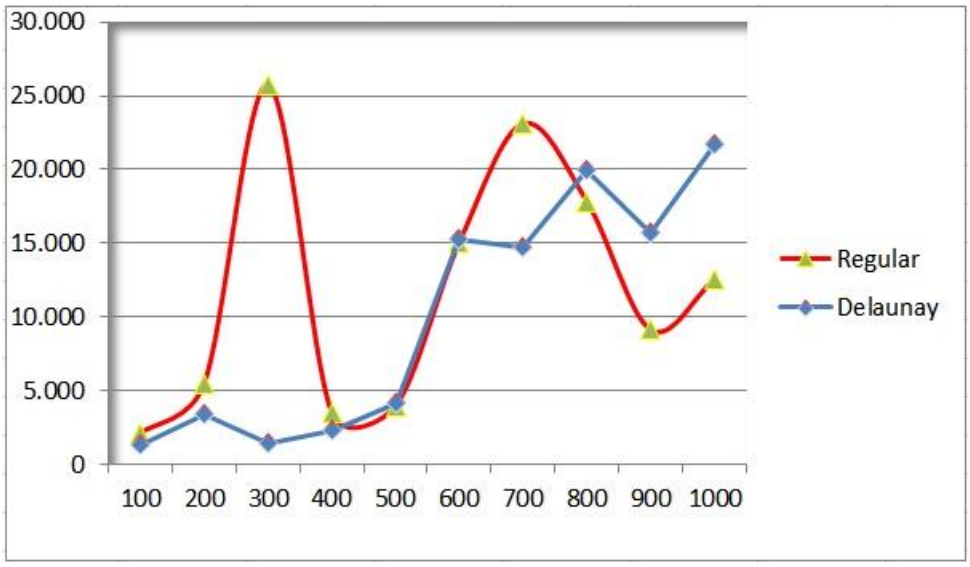
Κατά την εκτέλεση του κώδικα για το δεύτερο μέρος του τρίτου ερωτήματος αφήνουμε το χρήστη να επιλέγει ένα πλέγμα NXN μέσα στο οποίο θα παίρνει τιμές κάθε σημείο που δημιουργείται. Έτσι εύκολα μπορεί να παρατηρηθεί ότι για 100,200,…,1000 σημεία που δημιουργούνται αν αυτά δημιουργηθούν σε μικρό πλέγμα (π.χ 50X50) δηλαδή πολλές τιμές σε μικρό φάσμα συντεταγμένων τότε η κανονική τριγωνοποίηση είναι καλύτερη από την τριγωνοποίηση Delaunay. Αντίθετα, όταν έχουμε τα ίδια σημεία αλλά που το φάσμα τιμών τους είναι πολύ μεγαλύτερο όπως (π.χ. σε πλέγμα 1000X1000) τότε επειδή τα σημεία είναι πιο αραιά στο δισδιάστατο σύστημα συντεταγμένων η Delaunay τριγωνοποίηση είναι καλύτερη από την κανονική.

Για τα παραπάνω δύο, με τις σειρά που τα αναφέραμε πήραμε τα εξής 2 αποτελέσματα και σχεδιάσαμε τα εξής γραφήματα:

Αριθμός σημείων	**Χρόνος εκτέλεσης Regular**	**Χρόνος εκτέλεσης Delaunay**
100	**2.129 msec**	**1.305 msec**
200	**5.481 msec**	**3.341 msec**
300	**25.74 msec**	**1.390 msec**
400	**3.527 msec**	**3.264 msec**
500	**3.975 msec**	**4.128 msec**
600	**15.075msec**	**15.305msec**
700	**23.121msec**	**14.673msec**
800	**17.808msec**	**19.904msec**
900	**9.136 msec**	**15.744msec**

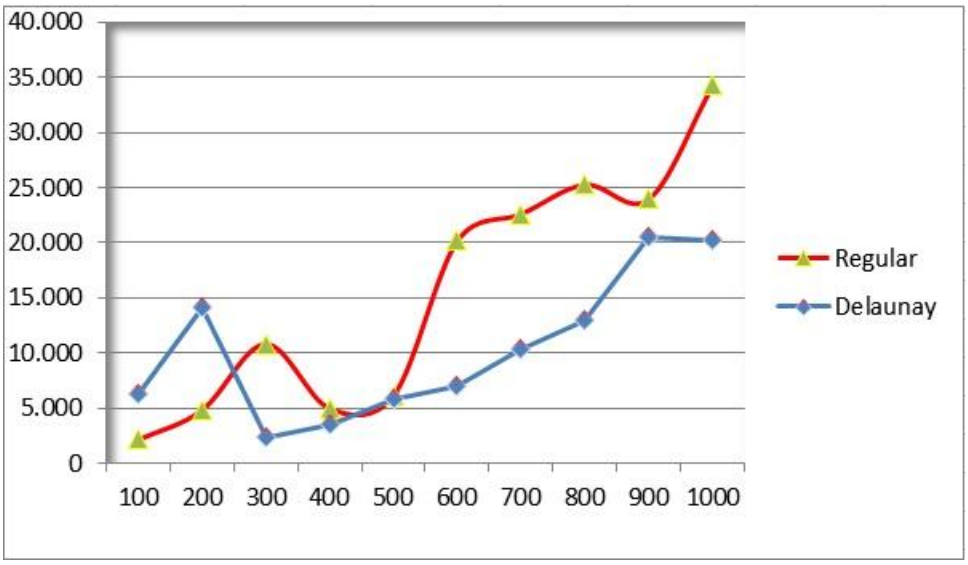
1000	**12.576msec**	**21.725msec**

Πίνακας 1: 50X50



Αριθμός σημείων	**Χρόνος εκτέλεσης Regular**	**Χρόνος εκτέλεσης Delaunay**
100	**2.128 msec**	**6.268 msec**
200	**4.807 msec**	**14.058 msec**
300	**10.804 msec**	**2.300 msec**
400	**4.962 msec**	**3.476 msec**
500	**6.112 msec**	**5.891 msec**
600	**20.141 msec**	**7.017 msec**
700	**22.587 msec**	**10.359 msec**
800	**25.249 msec**	**12.949 msec**
900	**23.935 msec**	**20.466 msec**
1000	**34.230 msec**	**20.219 msec**

Πίνακας 2: 1000X1000



Τέλος, αξίζει να σημειώσουμε ότι για το γράφημα όπου τα σημεία διαλέχτηκαν μέσα από το πλέγμα 50X50 τα vertices για την regular triangulation ήταν 1187 ενώ τα hidden vertices ήταν 4130! Ταυτόχρονα τα vertices για την delaunay triangulation ήταν 2215…

Ομοίως, για το γράφημα όπου τα σημεία διαλέχτηκαν μέσα από το πλέγμα 1000X1000 τα vertices για την regular triangulation ήταν 5484 ενώ τα hidden vertices ήταν μόλις 16! Ταυτόχρονα τα vertices για την delaunay triangulation ήταν 5485…