Υπολογιστική Γεωμετρία, Εαρινό εξάμηνο 2017-18

AM=1115201400140
ΠΑΠΑΓΕΩΡΓΙΟΥ ΒΑΣΙΛΕΙΟΣ-ΝΙΚΗΦΟΡΟΣ

Εργασία 1: Προγραμματιστική.

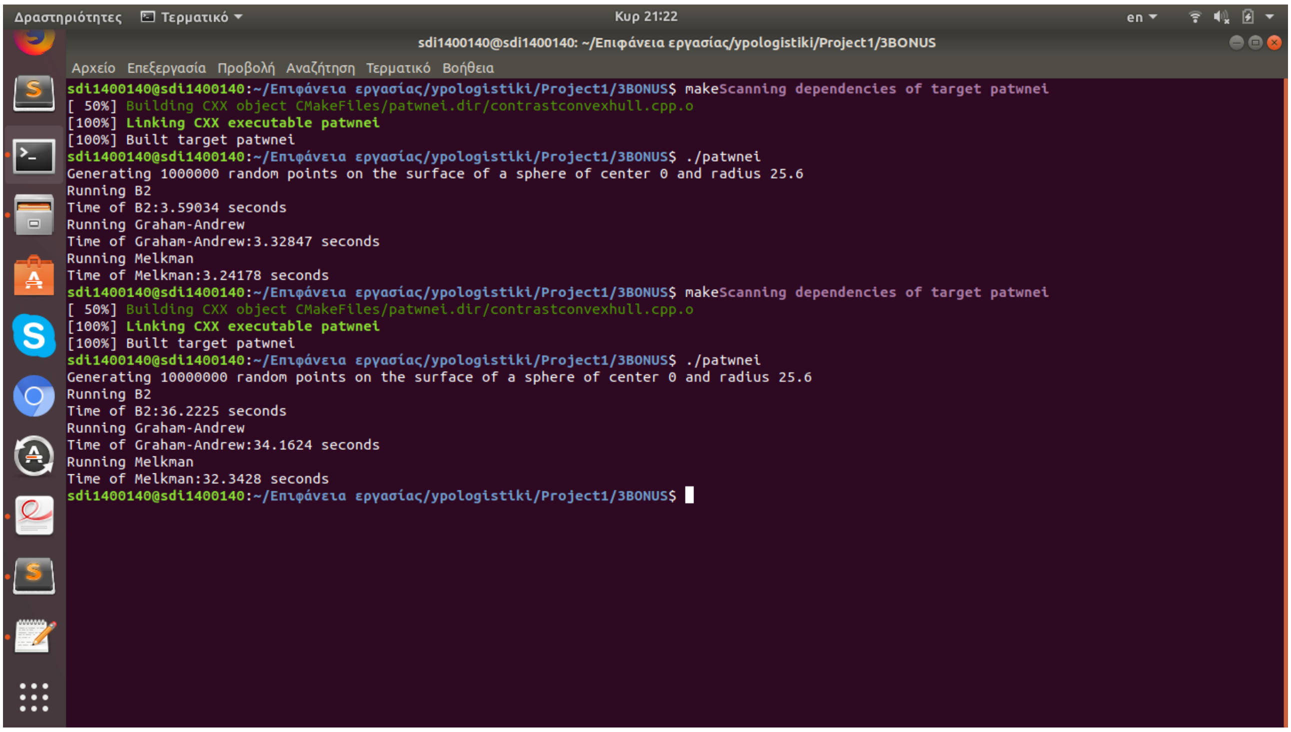
Στο ερώτημα 2Α(Convex_hull_1.cpp) υλοποιώ το ζητούμενο με χρήση της συνάρτησης της CGAL convex_hull_2,η οποία χρησιμοποιεί είτε τον αλγόριθμο του Bykat,είτε των Akl-Toussaint ανάλογα με την είσοδο.Η είσοδος γίνεται απ το stdin και η έξοδος εκτυπώνεται στο cout.

Στο ερώτημα 2Β(Convex_hull_2.cpp) υλοποιώ το ζητούμενο με χρήση της συνάρτησης της CGAL convex_hull_2,η οποία χρησιμοποιεί είτε τον αλγόριθμο του Bykat,είτε των Akl-Toussaint ανάλογα με την είσοδο.Η είσοδος γίνεται απ το αρχείο input1.txt που υπάρχει στον αντίστοιχο φάκελο και η έξοδος εκτυπώνεται στο output.txt(τα παραπάνω μέσω streams).

Στο ερώτημα 2Γ(Convex_hull_3.cpp) υλοποιώ το ζητούμενο με χρήση της συνάρτησης της CGAL convex_hull_2,η οποία χρησιμοποιεί είτε τον αλγόριθμο του Bykat,είτε των Akl-Toussaint ανάλογα με την είσοδο.Η είσοδος γίνεται απ το αρχείο input1.txt που υπάρχει στον αντίστοιχο φάκελο και η έξοδος εκτυπώνεται στο output.txt(τα παραπάνω μέσω streams).Το αρχείο input1.txt πρέπει να περιέχει τη λέξη Points ,πριν δοθούν οι συντεταγμένες των σημείων που θα ορίσουν την ευθεία,έτσι ώστε κατά το διάβασμά του να υπάρξει διαχωρισμός των σημείων για τα οποία θέλουμε το κυρτό περίβλημα(temp.txt).Στη συνέχεια δημιουργούμε την ευθεία μέσω του τύπου Line_2 της CGAL. Αφού υπολογίσουμε το κυρτό περίβλημα και τοποθετήσουμε τις συντεταγμένες του στο output.txt,για κάθε ζεύγος σημείων δημιουργούμε το ευθύγραμμο τμήμα που τα ενώνει(τύπος Segment_2) και μλεσω της συνάρτησης intersection() ,ελέγχουμε αν υπάρχουν σημεία τομής μεταξύ της ευθείας και κάθε ευθυγράμμου τμήματος του περιβλήματος.Κρατάμε το πρώτο και το τελευταίο σημείο του περιβλήματος και ελέγχουμε και γι αυτά ετσι ώστε να κλείσει ο κύκλος.

Στο ερώτημα 3(sphereconvexhull.cpp) υλοποιώ το ζητούμενο με χρήση του Kernel της CGAL Random_points_on_sphere_3<Point_3>,η οποία δημιουργεί N σημεία(πχ N=10) στη περιφέρεια μιας σφαίρας ακτίνας r(πχ r=25.60),με κέντρο το (0,0,0) και τοποθετώ τα σημεία αυτά σε ενα vector<Point_3>.Στη συνέχεια μέσω stream δημιουργώ το αρχείο temp.txt,με τα N σημεία και μέσω της convex_hull_2 με επιπλέον όρισμα το Trait που δημιουργώ για την προβολή των σημείων στον yz,μέσω της CGAL::Projection_traits_yz_3 ,παράγω το αρχείο output.txt,που περιέχει τα σημεία στον τρισδιάστατο χώρο που αντιστοιχούν στις κορυφές του 2διάστατου κυρτού περιβλήματος.

Στο ερώτημα 3-BONUS(contrastconvexhull.cpp),κάνω ακριβώς την ίδια διαδικασία με το ερώτημα 3 απλά υπολογίζω το κυρτό περίβλημα με τρεις διαφορετικούς αλγορίθμους(Bykat,Graham-Andrew,Melkman).Για N=10000000 παίρνω τους παρακάτω χρόνους:



Το παραπάνω είναι απόλυτα αιτιολογήσιμο,αν αναλογιστούμε τις πολυπλοκότητες καθενός απ τους αλγορίθμους.
Πιο συγκεκριμένα ο Bykat εχει O(nH),με n τις κορυφές της εισόδου και H τις κορυφές του περιβλήματος,για το οποίο παρατηρούμρ οτι για μικρα n ,είναι πολύ μικρό το H ,άρα η πολυπλοκότητα προσεγγίζει το O(n),ενώ για μεγάλα n,είναι μεγάλο το H και άρα η πολ/τα προσεγγίζει το O(n^2).

O scan αλγόριθμος Graham-Andrew έχει πολ/τα O(nlogn),ενώ ο αλγόριθμος του Melkman εφαρμόζεται μόνο σε πολύγωνα και είναι γραμμικός(O(n)).

Συνεπώς στην περίπτωσή μας με N= 10000000,το αναμενόμενο αποτέλεσμα είναι πιο γρήγορος να είναι ο Melkman,μετα ο Graham-Andrew και μετά ο Bykat.