

# 2η Προγραμματιστική εργασία

Αντώνης Σκαρλάτος sdi1400184

## Βασικές οδηγίες

Το πρόγραμμα μπορεί να τρέξει με δύο τρόπους. Ο ένας τρόπος είναι να τρέξουμε την εντολή `./executable inputFile.txt` και ο άλλος να τρέξουμε την εντολή `./executable -generate N`. Αν το τρέξουμε με το αρχείο, τότε το αρχείο θα πρέπει στην πρώτη γραμμή να έχει τον αριθμό των σημείων και στις υπόλοιπες γραμμές τα σημεία με τις συντεταγμένες τους (x, y, z) όπου οι συντεταγμένες χωρίζονται απλώς με ένα κενό.

## 2ο ερώτημα

Η συνάρτηση `beneath_beyond_3()` ορίζεται στο αρχείο `'beneath_beyond_3.h'`. Στην αρχή τα σημεία ταξινομούνται λεξικογραφικά. Στην συνέχεια δημιουργείται ένα αρχικό τετράεδρο και ακολουθείται η λογική του αλγορίθμου. Υπάρχουν κάποια βοηθητικά σχόλια στον κώδικα για διευκόλυνση.

## Σύγκριση αλγορίθμων

Ο αλγόριθμος που υλοποιείται στο 2ο ερώτημα είναι πιο αργός από τον αλγόριθμο του 1ου ερωτήματος. Ο λόγος είναι ότι στο 2ο ερώτημα, κάθε φορά που κάνουμε εισαγωγή μια νέα κορυφή, πρέπει να διατρέξουμε όλες τις έδρες για να δούμε ποιες είναι κόκκινες. Αυτό σημαίνει ότι θέλουμε  $O(n)$  χρόνο για να βρούμε τις κόκκινες έδρες στο 2ο ερώτημα, ενώ αυτό γίνεται σε χρόνο  $O(\log n)$  στο 1ο ερώτημα. Ακόμα ο τρόπος που δουλεύει ο `incremental_builder` στο 2ο ερώτημα μας αναγκάζει να διατρέχουμε

πάλι όλες τις κορυφές για να βρούμε το καινούργιο τους indexing(μέσω του map<>), ώστε να ξέρουμε πως πρέπει να δημιουργήσουμε τα καινούργια facets. Τελικά πέρα από τον παράγοντα του mc-Mullen, στο 1ο ερώτημα έχουμε τον παράγοντα  $O(n \cdot \log n)$ , ενώ στο 2ο ερώτημα έχουμε τον παράγοντα  $O(n^2)$ .

	$N = 100$	$N = 1000$	$N = 10000$
CH 1ου ερωτήματος	0.004sec	0.24sec	0.084sec
CH 2ου ερωτήματος	0.004sec	0.84sec	2.116sec