

# Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Χειμερινό εξάμηνο 2022-23

## 1<sup>η</sup> Προγραμματιστική Εργασία

Πολυγωνοποίηση σημειοσυνόλου με τη χρήση της βιβλιοθήκης CGAL (C++)

Η άσκηση θα υλοποιηθεί σε σύστημα Linux και θα υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 4/11 στις 23.59. Ορισμένα ερωτήματα είναι υποχρεωτικά μόνο για ομάδες 3 ατομών.

### Περιγραφή της εργασίας

0. Εγκατάσταση CGAL 5.5

1. Υλοποίηση 2 (εκ των 3) αλγορίθμων για την πολυγωνοποίηση σημειοσυνόλου  $S$  το οποίο περιέχει  $n$  σημεία στον χώρο  $\mathbb{R}^2$ . Οι τριάδες υλοποιούν και τους 3 αλγόριθμους.

ΑΥΞΗΤΙΚΟΣ. (a) Ταξινόμηση σημείων ως προς μια συντεταγμένη και επιλογή αρχικού τριγώνου. (b) Επαναληπτική διαδικασία προσθήκης σημείου στην πολυγωνική γραμμή  $A$ : (b1) Εύρεση κόκκινων ακμών στο ΚΠ. (b2) Για κάθε κόκκινη ακμή, απαρίθμηση ορατών ακμών στην  $A$ , b3. Επιλογή ορατής ακμής.

ΜΕ ΒΑΣΗ ΤΟ ΚΠ. (a) Υπολογισμός ΚΠ και αρχικοποίηση της πολυγωνικής γραμμής  $A$ . (β). Για κάθε ακμή της  $A$  εύρεση πλησιέστερου εσωτερικού σημείου, επιλογή ακμής και επέκταση της  $A$  ώστε να περιλάβει το σημείο (χωρίς να καθιστά εξωτερικό κάποιο άλλο). Επανάληψη της διαδικασίας μέχρι το σύνολο των εσωτερικών σημείων να μην έχει σημεία.

ΚΑΤΑΣΚΕΥΗ Επάλληλων ΚΠ (Κρεμμύδι). (α) Κατασκευή ΚΠ (βάθους 0), αφαίρεση κορυφών του από το  $S$ , κατασκευή ΚΠ (βάθους 1), αφαίρεση κορυφών, ΚΠ (βάθους 2) κοκ, μέχρι να μείνουν 0, 1 ή 2 εσωτερικά σημεία. (β) Επαναληπτική διαδικασία για κάθε ΚΠ βάθους  $i$ : (β1) Αφαίρεση ακμής  $i(m), i(m+1)$ , (β2) Εύρεση κοντινότερου σημείου  $(i+1)(k)$  στο ΚΠ βάθους  $i+1$ . (β3) Αφαίρεση ορατής ακμής  $(i+1)(k), (i+1)(\lambda)$ . (β4) Σύνδεση κορυφών  $(i+1)(k), i(m)$  και  $i(m+1), (i+1)(\lambda)$ . (β5) Ανάθεση κορυφών ακμής για αφαίρεση στο βάθος  $i+1$ ,  $(i+1)(m)$  και  $(i+1)(m+1)$  προχωρώντας δεξιόστροφα από το  $(i+1)(k)$  και αριστερόστροφα από το  $(i+1)(\lambda)$  μέχρι να φτάσουμε σε γειτονικές κορυφές.

2. Υπολογισμός επιφάνειας απλού πολυγώνου (π.χ. με τη συνάρτηση της CGAL ή κατά τη διάρκεια κατασκευής της  $A$ ). Μόνο για τριάδες: Υλοποίηση του αλγόριθμου Pick για τον υπολογισμό επιφάνειας απλού πολυγώνου με κορυφές σημεία ακέραιων συντεταγμένων.

3. Επιλογή κριτηρίων για τα βήματα των 3 αλγορίθμων και σύγκριση ως προς την επιφάνεια των απλών πολυγώνων που κατασκευάζονται.

### Αρχικοποίηση αυξητικού αλγόριθμου

1. Ταξινόμηση ως προς  $x$  (a) φθίνουσα, (b) αύξουσα 2. Ταξινόμηση ως προς  $y$  (a) φθίνουσα, (b) αύξουσα

### Επιλογή ορατής ακμής

1. Τυχαία επιλογή ορατής ακμής
2. Επιλογή ορατής ακμής έτσι ώστε να προστίθεται ελάχιστο εμβαδό
3. Επιλογή ορατής ακμής έτσι ώστε να προστίθεται μέγιστο εμβαδό.

## Αρχικοποίηση αναδρομικής κατασκευής ΚΠ

1. Τυχαία επιλογή. 2. Επιλογή σημείων με μικρότερο  $x$  3. Επιλογή σημείων με μεγαλύτερο  $x$  4. Επιλογή σημείων με μικρότερο  $y$  5. Επιλογή σημείων με μεγαλύτερο  $y$

## ΕΙΣΟΔΟΣ

Ένα αρχείο κειμένου για την είσοδο του **S** διαχωρισμένο με σπηλοθέτες (tab-separated), με την ακόλουθη γραμμογράφηση:

```
# <γραμμή περιγραφής σημειοσυνόλου>
# parameters "convex_hull": {"area": "X"} // όπου X η επιφάνεια του ΚΠ2
0      x0      y0
1      x1      y1
...    ...    ...
n-1    xn      yn
```

όπου  $n$  είναι το πλήθος των σημείων του σημειοσυνόλου και  $x_i, y_i$  οι συντεταγμένες (θετικοί ακέραιοι).

Το αρχείο εισόδου, το αρχείο εξόδου και οι παράμετροι του αλγόριθμου δίνονται ως παράμετροι γραμμής εντολών. Η εκτέλεση γίνεται μέσω της εντολής:

```
./to_polygon -i <point set input file> -o <output file> -algorithm <incremental or convex_hull or onion> -edge_selection <1 or 2 or 3 | όχι στο onion> -initialization <1a or 1b or 2a or 2b | μόνο στον αυξητικό αλγόριθμο> -onion_initialization <1 to 5>
```

## ΕΞΟΔΟΣ

Αρχείο κειμένου που περιέχει τις κορυφές και τις ακμές του πολυγώνου, την επιφάνειά του και το λόγο της επιφάνειάς του προς την επιφάνεια του κυρτού του περιβλήματος.

Π.χ. για τρίγωνο με κορυφές τις (0,0), (4,0), (4,4)

```
Polygonization
0 0
4 0
4 4
0 0 4 0
4 0 4 4
4 4 0 0
Algorithm: <incremental or convex_hull or onion>_edge_selection<1 or 2 or 3
where applicable>_initialization<incremental and onion options accordingly>
area: 8
pick_calculated_area: 8 // μόνο για ομάδες 3 ατόμων
ratio: 1
construction time: <milliseconds as integer>
```

Η αναφορά που θα παραδοθεί θα πρέπει να περιλαμβάνει σχολιασμό των αποτελεσμάτων για σημειοσύνολα διαφορετικού μεγέθους και είδους (ομοιόμορφα τυχαία και από δειγματοληψία εικόνων στο eclass) καθώς και για διαφορετικά κριτήρια επιλογής ακμών και αρχικοποίησης ως προς την επιφάνεια των πολυγώνων που κατασκευάζονται.

## Επιπρόσθετες απαιτήσεις

- 1) Το πρόγραμμα πρέπει να είναι καλά οργανωμένο με χωρισμό των δηλώσεων / ορισμών των συναρτήσεων, των δομών και των τύπων δεδομένων σε λογικές ομάδες που αντιστοιχούν σε ξεχωριστά αρχεία επικεφαλίδων και πηγαίου κώδικα. Βαθμολογείται και η ποιότητα του κώδικα (π.χ. αποφυγή memory leaks). Η μεταγλώττιση του προγράμματος πρέπει να γίνεται με τη χρήση του εργαλείου make και των εργαλείων δημιουργίας cmake scripts που παρέχει η βιβλιοθήκη CGAL.
- 2) Το παραδοτέο πρέπει να είναι επαρκώς τεκμηριωμένο με πλήρη σχολιασμό του κώδικα και την ύπαρξη αρχείου readme το οποίο περιλαμβάνει κατ'ελάχιστο: α) τίτλο και περιγραφή του προγράμματος, β) κατάλογο των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους, γ) οδηγίες μεταγλώττισης του προγράμματος, δ) οδηγίες χρήσης του προγράμματος και ε) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.
- 3) Η υλοποίηση του προγράμματος θα πρέπει να γίνει με την χρήση συστήματος διαχείρισης εκδόσεων λογισμικού και συνεργασίας Git.

## Χειμερινό εξάμηνο 2022-23

### 2<sup>η</sup> Προγραμματιστική Εργασία

#### Πολυγωνοποίηση σημειοσυνόλου βέλτιστης επιφάνειας με τη χρήση της βιβλιοθήκης CGAL (C++)

Η άσκηση θα υλοποιηθεί σε σύστημα Linux και θα υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 9/12 στις 23.59. Ορισμένα ερωτήματα είναι υποχρεωτικά μόνο για ομάδες 3 ατόμων.

#### Περιγραφή της εργασίας

1. Υλοποίηση των δύο πρώτων αλγορίθμων για την πολυγωνοποίηση βέλτιστης επιφάνειας σημειοσυνόλου **S** το οποίο περιέχει **n** σημεία στον χώρο  $\mathbb{R}^2$ . Οι τριάδες υλοποιούν και τον 3ο αλγόριθμο (Αποικία μυρμηγκιών).

ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ. (a) Επιλογή αλγορίθμου για την αρχική πολυγωνοποίηση σημειοσυνόλου από την 1<sup>η</sup> εργασία. (b) Εύρεση δυνατών τμημάτων της πολυγωνικής γραμμής  $V$ , μέγιστου πλήθους σημείων  $L$  ( $L \leq 10$ ), που μπορούν να συνδεθούν με κάθε ακμή του πολυγώνου, και επιλογή ενός εξ αυτών για τη δημιουργία νέου απλού πολυγώνου με μέγιστη ή ελάχιστη διαφορά εμβαδού. (c) Επανάληψη της διαδικασίας μέχρι η βελτίωση να είναι αμελητέα.

ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ [SIMULATED ANNEALING]. (a) Επιλογή αλγορίθμου για την αρχική πολυγωνοποίηση σημειοσυνόλου από την 1<sup>η</sup> εργασία. Υπολογισμός «ενέργειας» αρχικής κατάστασης. (b) Μετάβαση σε νέα κατάσταση με i) τοπικό βήμα αλλαγής σειράς μεταξύ δύο σημείων στην πολυγωνική αλυσίδα χωρίς να παραβιάζεται η απλότητα της πολυγωνικής γραμμής. Για τον έλεγχο εγκυρότητας της αλλαγής χρησιμοποιείται η δομή kd-Tree της CGAL. ii) καθολικό βήμα αλλαγής βάσει τυχαίων σημείων  $q, s$  της αλυσίδας. Αν  $p, q$  και  $s, t$  είναι ακμές, συνδέονται τα σημεία  $p$  και  $r$  και στη συνέχεια το  $q$  εισάγεται μεταξύ  $s$  και  $t$ . Για τον έλεγχο εγκυρότητας της μετάβασης πρέπει να ελεγχθεί εξαντλητικά ότι οι νέες ακμές δεν τέμνουν καμία άλλη ακμή της πολυγωνικής γραμμής. (c) Υπολογισμός της διαφοράς της ενέργειας της νέας κατάστασης (απαιτείται μόνο ο υπολογισμός της επιφάνειας των τριγώνων που προστίθενται και αφαιρούνται). (d) Αποδοχή ή απόρριψη της μετάβασης βάσει του κριτηρίου Metropolis  $e^{(-\Delta E/T)} > R(0,1)$  [τυχαίος θετικός αριθμός  $< 1$ ]. Στην αρχική κατάσταση η θερμοκρασία ορίζεται ίση με 1 και μειώνεται σε κάθε βήμα κατά  $1/L$ , όπου  $L$  το μέγιστο πλήθος επαναλήψεων.

Τα σημειοσύνολα που περιέχουν  $n > 1.000$  σημεία χωρίζονται σε  $\lceil (n-1)/(m-1) \rceil$  υποσύνολα που περιέχουν περίπου  $m$  σημεία το καθένα, βάσει της ταξινόμησης των σημείων ως προς τη συντεταγμένη  $x$  κατά αύξουσα σειρά ( $10 \leq m \leq 100$ ). Τα διαδοχικά υποσύνολα περιλαμβάνουν ένα κοινό σημείο και πρέπει να διασφαλιστεί ότι η δεξιότερη ακμή του κάτω περιβλήματος του κάθε υποσυνόλου είναι γνήσια αύξουσα ως προς την τεταγμένη, ενώ η αριστερότερη ακμή του κάτω περιβλήματος του κάθε υποσυνόλου είναι γνήσια φθίνουσα ως προς την τεταγμένη. Η διαδικασία κατάτμησης περιγράφεται αναλυτικά στη δημοσίευση των Goren et al. (2b στο eClass, σ. 6). Οι δεξιότερες και οι αριστερότερες ακμές των κάτω περιβλημάτων των υποσυνόλων μαρκάρονται κατάλληλα, εκτός από την αριστερότερη ακμή του κάτω περιβλήματος πρώτου υποσυνόλου και τη δεξιότερη ακμή του τελευταίου. Στη συνέχεια επιλύεται το πρόβλημα πολυγωνοποίησης του κάθε υποσυνόλου με καθολικό βήμα αλλαγής υπό τον περιορισμό ότι οι μαρκαρισμένες ακμές περιλαμβάνονται στα υποπολύγωνα που δημιουργούνται. Αυτό επιτυγχάνεται με τη συμπερίληψη αυτών των ακμών στην αρχική κατάσταση και με την μη πραγματοποίηση βημάτων αλλαγής που να παραβιάζουν αυτόν τον περιορισμό. Τελικά, οι λύσεις των υποπροβλημάτων συνενώνονται. Αν  $p, q$  και  $r$  είναι οι μαρκαρισμένες ακμές των υποσυνόλων  $S_i$  και  $S_{i+1}$ , αυτές αφαιρούνται και εισάγεται η ακμή  $p, r$ .

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΣΩ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ. (a) Τροποποίηση αυξητικού αλγορίθμου έτσι ώστε η αρχική επιλογή τριγώνου να είναι τυχαία, τα σημεία να μην είναι ταξινομημένα και η επιλογή του επόμενου εφικτού σημείου προς εισαγωγή να είναι επίσης τυχαία βάσει της πιθανότητας που ορίζει ο αλγόριθμος της αποικίας μυρμηγκιών. Αξιολόγηση λύσης βάσει λόγου επιφάνειας πολυγώνου προς επιφάνεια ΚΠ και αποθήκευση τρέχουσας καλύτερης λύσης (b) Ανανέωση μονοπατιών φερομόνης των ακμών (μείωση λόγω χρονικής εξασθένισης («εξάτμιση») και αύξηση λόγω επιλογής ακμής. (c) Επανάληψη διαδικασίας για  $L$  κύκλους και επιστροφή καλύτερης λύσης. Η ευριστική αξία των ακμών είναι ίση με την ευκλείδεια απόσταση των κορυφών τους αν πρόκειται για πρόβλημα μεγιστοποίησης της επιφάνειας και με το αντίστροφό της αν πρόκειται για πρόβλημα ελαχιστοποίησης της επιφάνειας.

## ΕΙΣΟΔΟΣ

Ένα αρχείο κειμένου για την είσοδο του **S** διαχωρισμένο με στηλοθέτες (tab-separated), με την ακόλουθη γραμμογράφηση:

```
# <γραμμή περιγραφής σημειοσυνόλου>
# parameters "convex_hull": {"area": "X"} // όπου X η επιφάνεια του ΚΠ2
0      x0      y0
1      x1      y1
...    ...    ...
n-1    xn      yn
```

όπου  $n$  είναι το πλήθος των σημείων του σημειοσυνόλου και  $x_i, y_i$  οι συντεταγμένες (θετικοί ακέραιοι).

Το αρχείο εισόδου, το αρχείο εξόδου και οι παράμετροι του αλγορίθμου δίνονται ως παράμετροι γραμμής εντολών. Η εκτέλεση γίνεται μέσω της εντολής:

```
./optimal_polygon -i <point set input file> -o <output file> -algorithm
<local_search or simulated_annealing or ant_colony> -L [L parameter according
to algorithm] -max [maximal area polygonization] -min [minimal area
polygonization] -threshold <double> [in local search] -annealing <"local" or
"global" or "subdivision" in simulated annealing>
```

Επιλέγεται μία μόνο επιλογή εκ των  $\max$  και  $\min$ . Στην περίπτωση του αλγορίθμου αποικίας μυρμηγκιών δίνονται στη γραμμή εντολών και οι παράμετροι  $\alpha, \beta, \rho$  ως δεκαδικοί αριθμοί καθώς και η παράμετρος  $\text{elitism}$  που δέχεται τις τιμές 1 και 0.

## ΕΞΟΔΟΣ

Αρχείο κειμένου που περιέχει τις κορυφές και τις ακμές του βέλτιστου πολυγώνου, την επιφάνειά του και το λόγο της επιφάνειάς του προς την επιφάνεια του κυρτού του περιβλήματος. Εκτυπώνεται επίσης η επιφάνεια και ο λόγος του αρχικού πολυγώνου στους 2 πρώτους αλγόριθμους.

```
Optimal Area Polygonization
```

```
0 0
```

```
4 0
```

```
4 4
```

```
...
```

```
0 0 4 0
```

```
...
```

```
Algorithm: <local_search or simulated_annealing or ant_colony>_[max or min]
```

```
area_initial <double> [only in local search and simulated annealing]
```

```
area: <double>
```

```
ratio_initial <double> [only in local search and simulated annealing]
```

```
ratio: <double>
```

```
construction time: <milliseconds as integer>
```

Η αναφορά που θα παραδοθεί θα πρέπει να περιλαμβάνει σύγκριση και σχολιασμό των αποτελεσμάτων για σημειοσύνολα διαφορετικού μεγέθους και είδους (ομοιόμορφα τυχαία και από δειγματοληψία εικόνων στο eclass) ως προς την απόδοση χρόνου και την επίτευξη της βέλτιστης λύσης.

## Επιπρόσθετες απαιτήσεις

- 1) Το πρόγραμμα πρέπει να είναι καλά οργανωμένο με χωρισμό των δηλώσεων / ορισμών των συναρτήσεων, των δομών και των τύπων δεδομένων σε λογικές ομάδες που αντιστοιχούν σε ξεχωριστά αρχεία επικεφαλίδων και πηγαίου κώδικα. Βαθμολογείται και η ποιότητα του κώδικα (π.χ. αποφυγή memory leaks). Η μεταγλώττιση του προγράμματος πρέπει να γίνεται με τη χρήση του εργαλείου make και των εργαλείων δημιουργίας cmake scripts που παρέχει η βιβλιοθήκη CGAL.
- 2) Το παραδοτέο πρέπει να είναι επαρκώς τεκμηριωμένο με πλήρη σχολιασμό του κώδικα και την ύπαρξη αρχείου readme το οποίο περιλαμβάνει κατ' ελάχιστο: α) τίτλο και περιγραφή του προγράμματος, β) κατάλογο των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους, γ) οδηγίες μεταγλώττισης του προγράμματος, δ) οδηγίες χρήσης του προγράμματος και ε) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.
- 3) Η υλοποίηση του προγράμματος θα πρέπει να γίνει με την χρήση συστήματος διαχείρισης εκδόσεων λογισμικού και συνεργασίας Git.

Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Χειμερινό εξάμηνο 2022-23

### 3<sup>η</sup> Προγραμματιστική Εργασία

Βελτιστοποίηση πολυγωνοποίησης σημειοσυνόλου βέλτιστης επιφάνειας, ανάπτυξη εφαρμογής για τη συγκριτική αξιολόγηση των αλγόριθμων πολυγωνοποίησης και διαγωνισμός.

Η άσκηση θα υλοποιηθεί σε σύστημα Linux και θα υποβληθεί στις Εργασίες του e-class το αργότερο την Τετάρτη 11/01/23 στις 23.59.

#### Περιγραφή της εργασίας

Το πρώτο ζητούμενο είναι η ολοκλήρωση και η βελτιστοποίηση της υλοποίησης όλων των αλγόριθμων της 1<sup>ης</sup> και της 2<sup>ης</sup> εργασίας ως προς την **ταχύτητα** και την **ακρίβεια** των προσεγγίσεων. Εν συνεχεία, ζητείται η ανάπτυξη εφαρμογής για τη συγκριτική αξιολόγηση των αλγόριθμων πολυγωνοποίησης βέλτιστης επιφάνειας της 2<sup>ης</sup> εργασίας. Η εφαρμογή θα δέχεται ως παράμετρο ένα κατάλογο με σημειοσύνολα που θα χρησιμοποιηθούν για τη συγκριτική αξιολόγηση.

Η εφαρμογή συγκρίνει όλες τις παραλλαγές των αλγόριθμων συμπεριλαμβανομένου του βήματος αρχικοποίησης για την εύρεση του αρχικού πολυγώνου και μπορεί να χρησιμοποιήσει σύνθετες ευριστικές τεχνικές **συνδυάζοντας** τις υλοποιήσεις που έχετε. Οι παράμετροι των αλγόριθμων (π.χ. L, ρ) επιλέγονται αυτόματα μέσω ενός σταδίου προεπεξεργασίας. Εναλλακτικά, χρησιμοποιούνται προεπιλεγμένες τιμές. Τα αποτελέσματα θα πρέπει να παράγονται εντός μιας χρονικής διάρκειας **cut-off** που ορίζεται για σημειοσύνολα πλήθους n σημείων σε f(n) milliseconds στο μηχάνημα linux30 και βάσει της συνάρτησης  $f(n) = 500 * n$  msec. Τα αποτελέσματα στα οποία συγκρίνονται οι εφαρμογές εκτυπώνονται σε αρχείο ως εξής:

(Score\_min) Το άθροισμα των ελάχιστων επιμέρους σκορ για πολύγωνα ελάχιστης επιφάνειας

(Score\_max) Το άθροισμα των μέγιστων επιμέρους σκορ για πολύγωνα μέγιστης επιφάνειας

(Bound\_min) Άνω φράγμα των επιμέρους σκορ για την ελαχιστοποίηση

(Bound\_max) Κάτω φράγμα των επιμέρους σκορ για τη μεγιστοποίηση

Για κάθε σημειοσύνολο, το επιμέρους σκορ είναι ο λόγος μεταξύ του επιτευχθέντος εμβαδού προς το εμβαδόν του κυρτού περιβλήματος, δηλαδή ένας αριθμός μεταξύ 0 και 1. Όταν δεν βρίσκεται εφικτή λύση ή εξαντλείται ο χρόνος εύρεσης λύσης cut-off, το σκορ είναι 1 (για ελαχιστοποίηση) ή 0 (για μεγιστοποίηση). Τυπώνεται ως φράγμα (bound) το χειρότερο επιμέρους σκορ ανά πρόβλημα.

Μετά το τέλος της διαδικασίας αξιολόγησης η εφαρμογή θα εξάγει τα αποτελέσματα σε αρχείο κειμένου με κατάλληλη μορφοποίηση. Ο διαγωνισμός αντιστοιχεί σε 10% επί του συνολικού βαθμού της εργασίας.

## ΕΙΣΟΔΟΣ

Ο κατάλογος με τα αρχεία κειμένου που περιλαμβάνουν τα σημειosύνολα προς πολυγωνοποίηση. Θα δοθούν συγκεκριμένα αρχεία. Η γραμμογράφηση των αρχείων που αντιστοιχούν στα σημειosύνολα ακολουθεί τις προδιαγραφές της 2<sup>ης</sup> εργασίας.

```
$./evaluate -i <point set path> -o <output file> -preprocess <optional>
```

## ΕΞΟΔΟΣ

Αρχείο κειμένου που περιλαμβάνει τα αποτελέσματα της αξιολόγησης

```
Size || <Algorithm 1> || <Algorithm 2> || . . .
Size || min score | max score | min bound | max bound || <same cols as previously>
10 || <double> | <double> | <double> | <double> || . . .
20 || . . .
30
40
50
60
70
80
90
100
200
400
800
1000
2000
5000
10000
100000
```

Η αναφορά που θα παραδοθεί πρέπει να περιλαμβάνει αναλυτικό σχολιασμό των αποτελεσμάτων της συγκριτικής αξιολόγησης των αλγόριθμων.

## Επιπρόσθετες απαιτήσεις

- 1) Το πρόγραμμα πρέπει να είναι καλά οργανωμένο με χωρισμό των δηλώσεων / ορισμών των συναρτήσεων, των δομών και των τύπων δεδομένων σε λογικές ομάδες που αντιστοιχούν σε ξεχωριστά αρχεία επικεφαλίδων και πηγαίου κώδικα. Βαθμολογείται και η ποιότητα του κώδικα (π.χ. αποφυγή memory leaks). Η μεταγλώττιση του προγράμματος πρέπει να γίνεται με τη χρήση του make και των εργαλείων δημιουργίας cmake scripts που παρέχει η βιβλιοθήκη CGAL.
- 2) Το παραδοτέο πρέπει να είναι επαρκώς τεκμηριωμένο με πλήρη σχολιασμό του κώδικα και την ύπαρξη readme το οποίο περιλαμβάνει κατ'ελάχιστο: α) τίτλο και περιγραφή του προγράμματος, β) κατάλογο των αρχείων κώδικα / επικεφαλίδων και περιγραφή τους, γ) οδηγίες μεταγλώττισης του προγράμματος, δ) οδηγίες χρήσης του προγράμματος, ε) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.
- 3) Η υλοποίηση του προγράμματος θα πρέπει να γίνει με την χρήση συστήματος διαχείρισης εκδόσεων λογισμικού και συνεργασίας Git.