

Ant colony

Δομη κωδικα/λειτουργιες:

Ο αλγοριθμος ειναι μια τροποποιηση του incremental algorithm απο την πρωτη εργασία σε συνδυασμο με τον ant colony algorithm.

Γενικη ιδεα:

Για την αποπειρα ευρεσης βελτιστου εμβαδου δημιουργειται ενας γραφος στον οποιο θα κινουνται τα μυρμήγκια που εχει σαν κομβους πολυγωνα. Αρχικα δημιουργουνται ολα τα τριγωνα που γινεται να υπαρχουν στο σημειοσυνολο. Αυτο γινεται με την συναρτηση generate3 η οποια διαλεγει 3 σημεια και βλεπει εαν δεν υπαρχει καποιο αλλο σημειο που να ειναι εσωτερικο στο τριγωνο. Εαν αυτο ισχυει το τριγωνο ειναι εγκυρο και μπαινει στην λιστα των πιθανων τριγωνων/εντασεται στον γραφο. Μετα για καθε μυρμήγκι επιλεγετα ενα αρχικο τριγωνο με βαση το ειδος του αλγοριθμου (εαν ειναι μεγιστοποιηση τοτε διαλεγει με βαση το ποσοστο του εμβαδου του συγκεκριμένου τριγωνου με το μεσο εμβαδο των υπολοιπων, αντιστοιχα για ελαχιστοποιηση). Επειτα αφου διαλεξει τριγωνω φτιαχνει ολα τα δυνατα τετραγωνα που ξεκινανε απο το τριγωνο. Αυτο γινεται με την συναρτηση GenerateX. Προσθετει αυτα τα τετραγωνα στο γραφο και σε καποιες δομες. Συνεχιζει μεχρι να τελειωσει το σημειοσυνολο και επειτα ξεκιναι νεο μυρμήγκι. Οταν τελειωσουν ολα τα μυρμηγκια τοτε βρισκεται το μεγαστο/ελαχιστο εμβαδον μεχρι τωρα και αποθηκευονται στην δομη paths τα μονοπατια των μυρμηγκιων. Καλειται η UpdtateTrails η οποια εαν βρει κομβο στον γραφο που το μυρμήγκι χρησιμοποίησε τοτε ανανεωνει καταλληλα την φερομονη. Επειτα “εξασθενειται” η φερομονη απο καθε ακμη και συνεχιζει στον επομενο κυκλο. Οταν τελειωσουν ολοι οι κυκλοι επιστρεφει το μεγαστο/ελαχιστο εμβαδον που βρηκε.

Δομες:

Γενικο: Τα πολυγωνα αντιστιχουνται με εναν αριθμο για να μην χρειαζεται να κτραταμε παντου την δομη των πολυγωνων.

polymap:Ενα map για να μπορούμε να φτιαχνουμε τα πολυγωνα με βαση τον αριθμο τους.

map:Ενα map για να μπορούμε να ξερούμε ποιο πολυγωνο εχει φτιαχτει μεχρι τωρα.

table1:Ο βασικος γραφος.

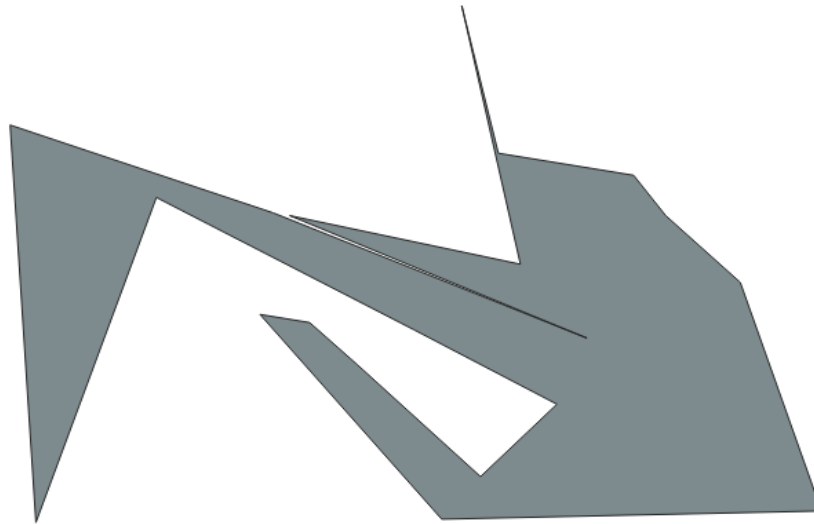
tables:Μια δομη η οποια περιεχει τον πατερα και τα παιδια ενος κομβου.

enumvals:Ενας πινακα που περιεχει τον αριθμο του πολυγωνου.

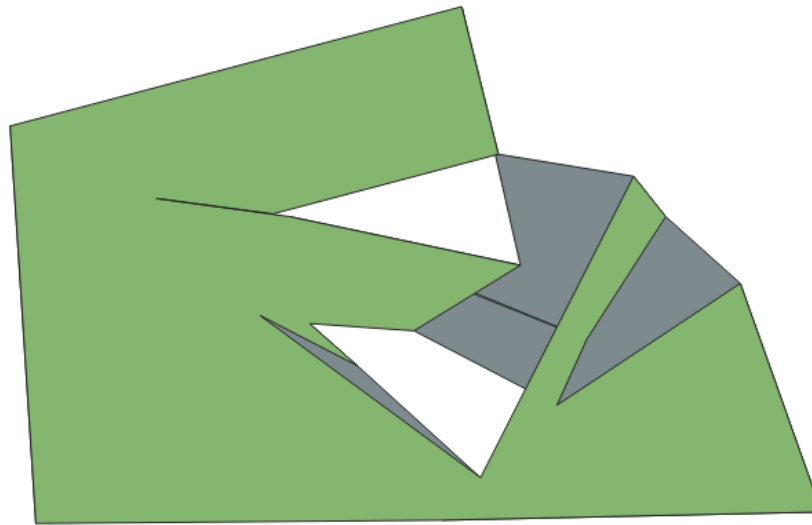
Παρατηρησεις:

Ο αλγοριθμος ειναι αρκετα αργος ,που ειναι λογικο , επειδη ειναι σε εναν βαθμο παρομοιος με brute force algorithm.Φτιαχνει καθε φορα ολα τα δυνατα πολυγωνα απο την συγγκεκριμενη θεση,εφοσον αυτα δεν εχουν φτιαχτει ηδη,που τοτε τα συνδεει στο γραφο.Παρακατω μπορουμε να δουμε καποιες εκτελεσεις του κωδικα για 15/20/50 σημεια.

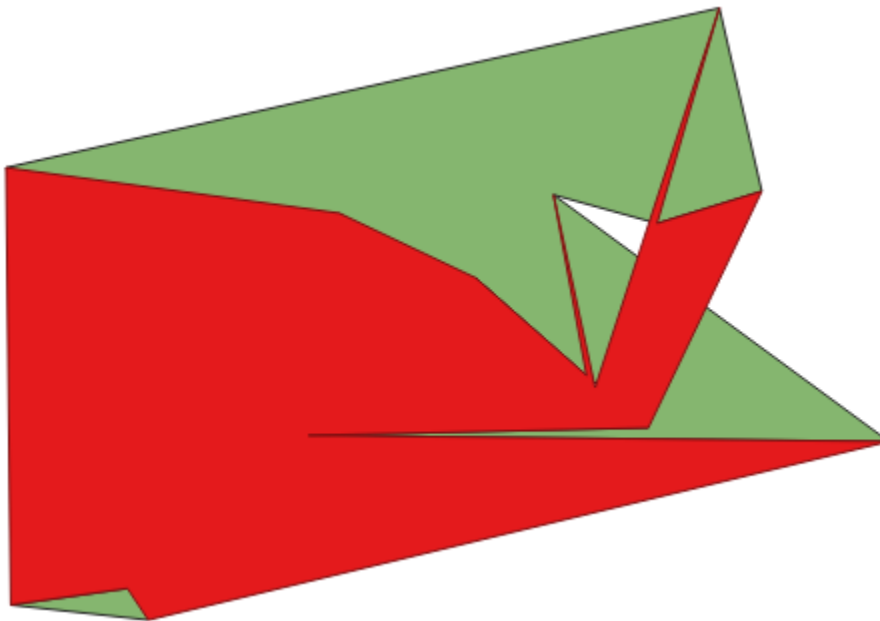
Εδω βλέπουμε το πολυγωνο που προεκηψε για 15 σημεια με $\alpha=1, \beta=1$ και $\Lambda=4$ για 15 σημεια.



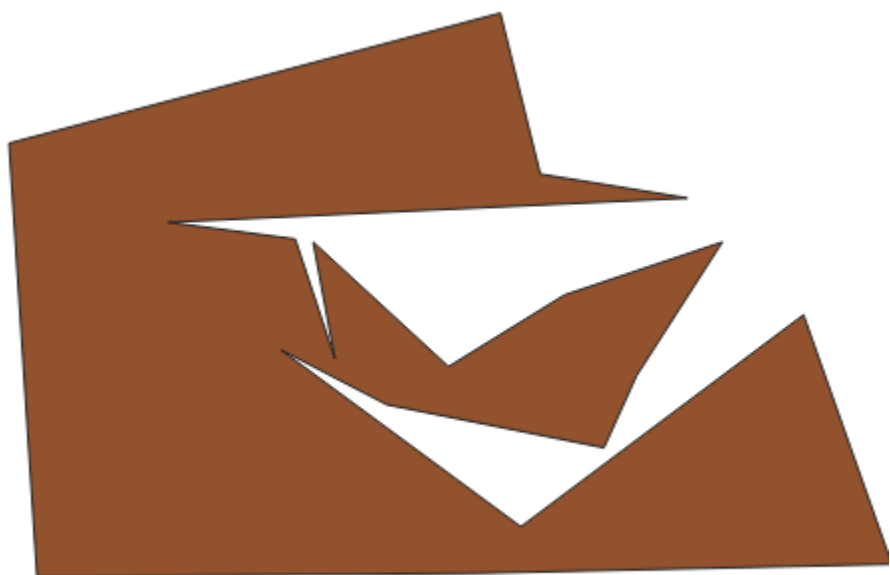
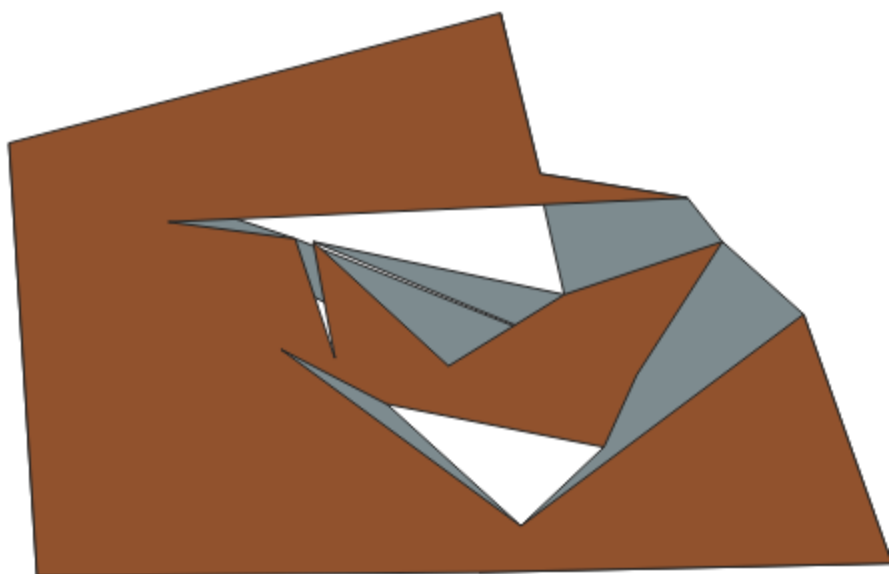
Και εδω και τα δυο πολυγωνα μαζι.

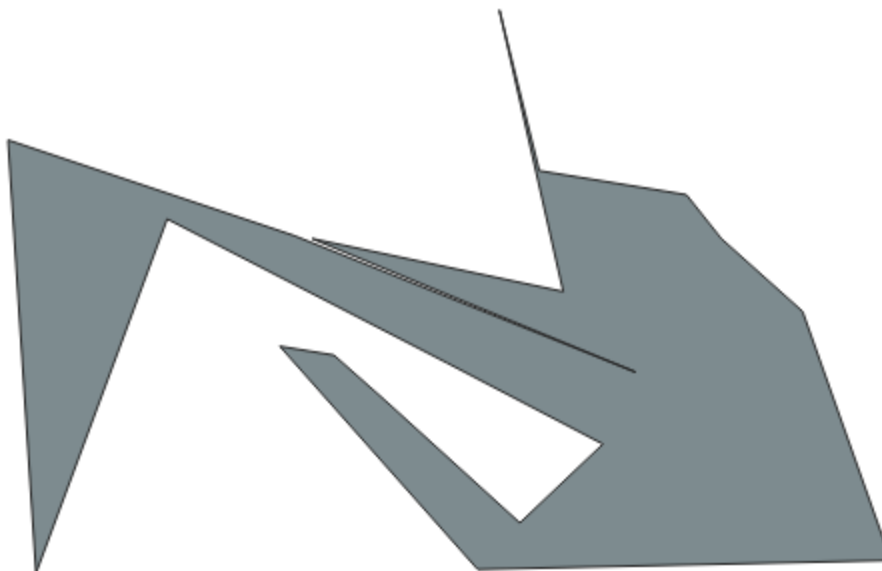


Επίσης για 15 σημεία.

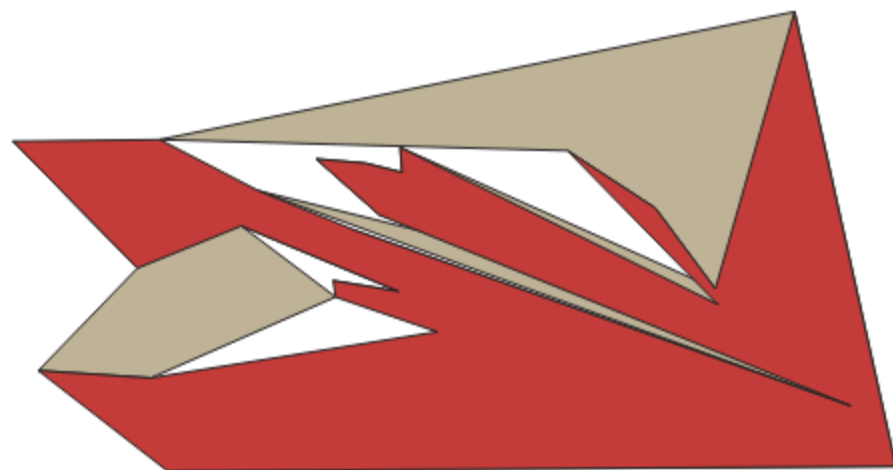


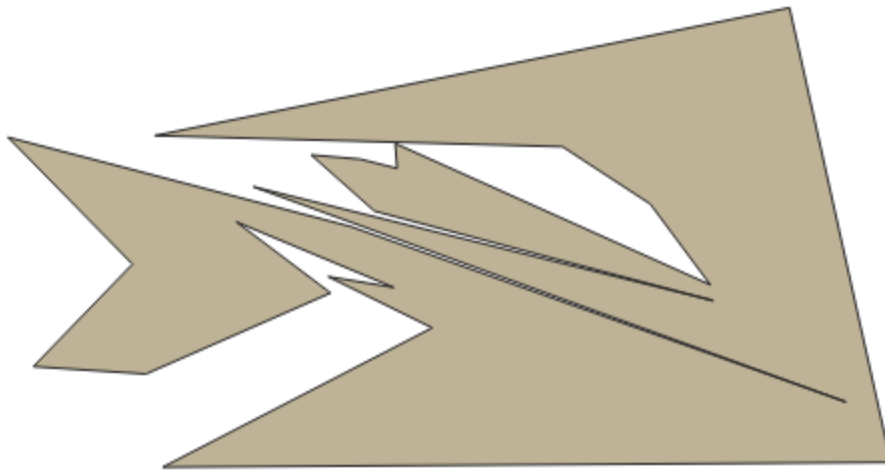
Εδώ βλέπουμε τα πολυγωνα για $\beta=3$ τώρα.



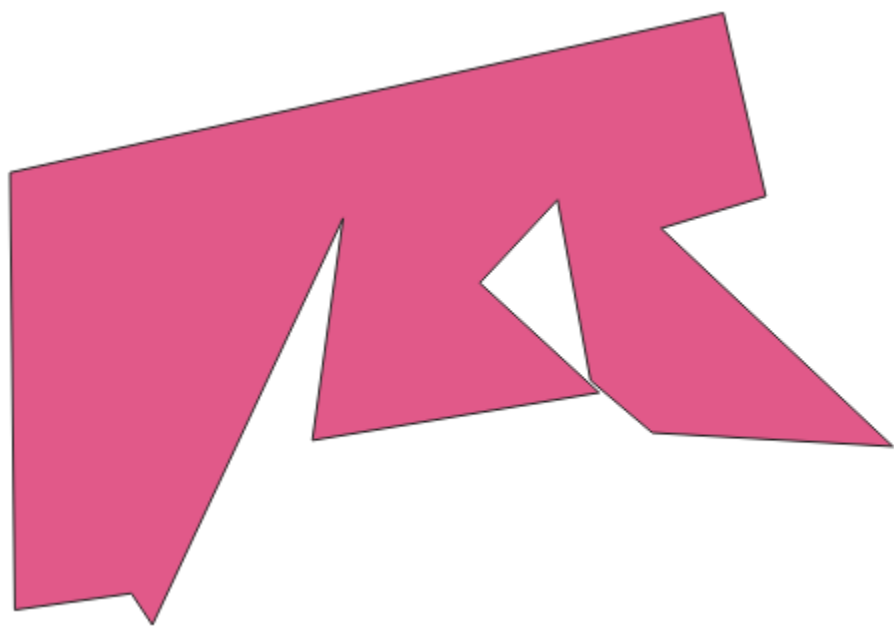
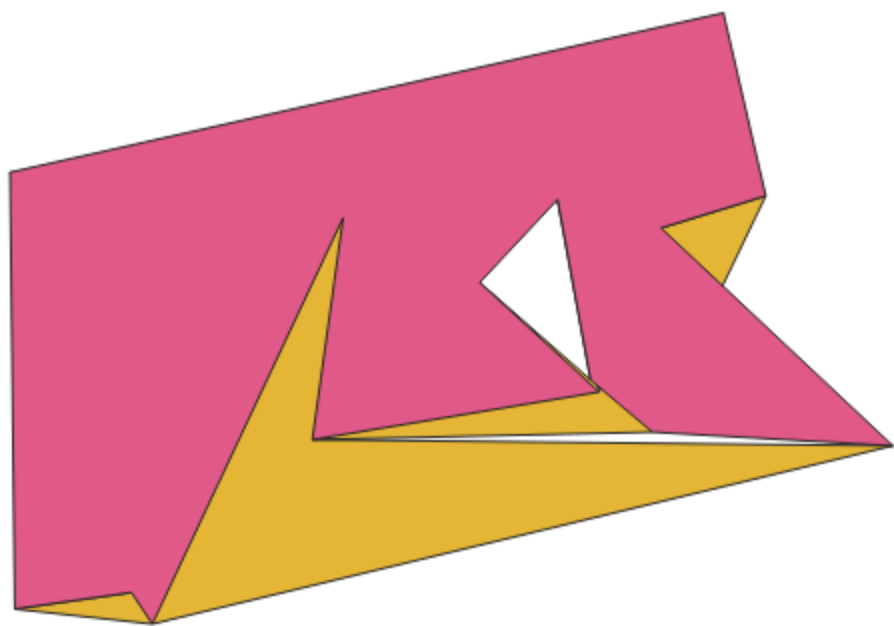


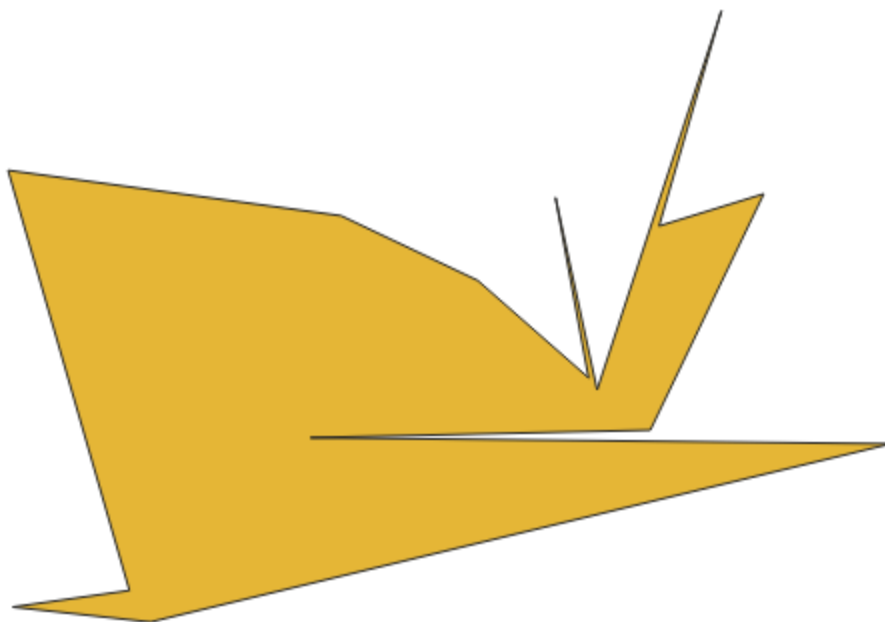
Και εδω για $\beta=3$ αλλα για 20 σημεια.





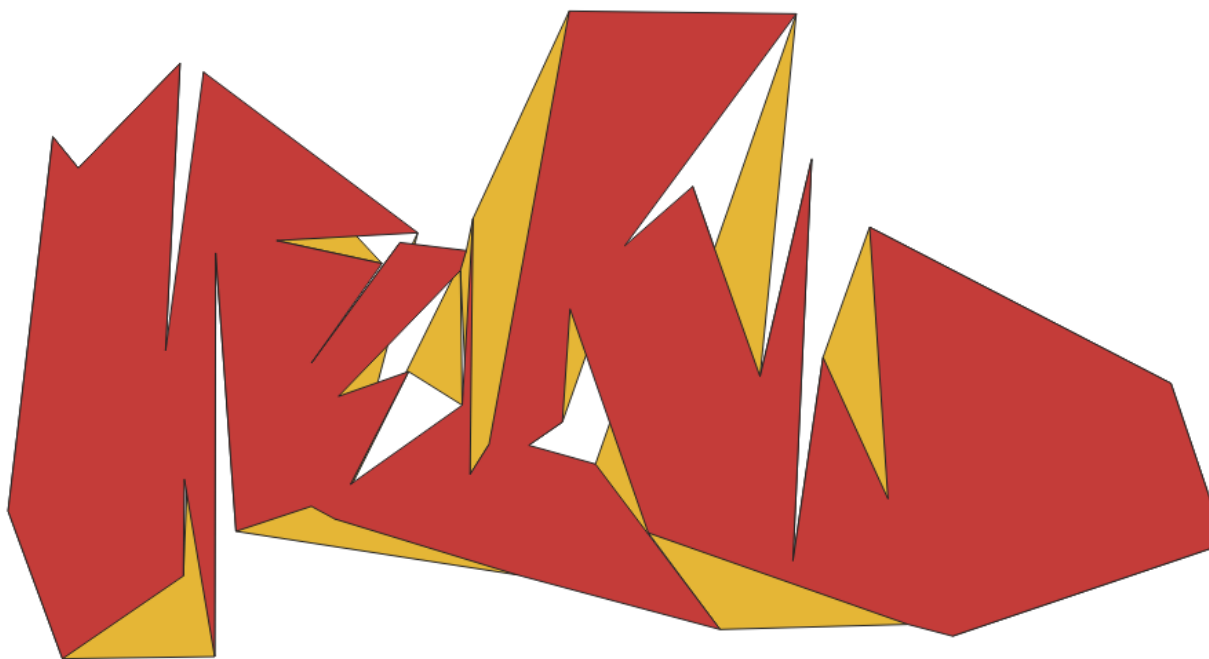
Τελος εδω παρατηρουμε τα πολυγωνα για $\beta=3$ και για $\Lambda=7$

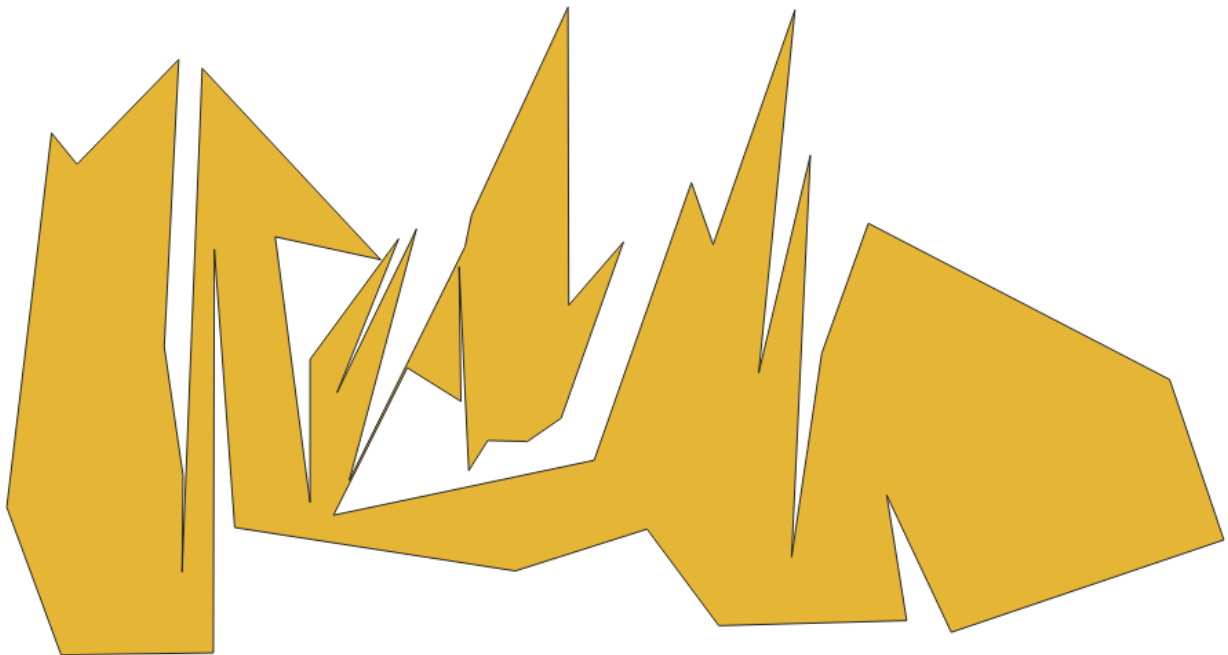




Μια παρατήρηση είναι ότι όσο ανεβαίνει το β τόσο μικραίνει η προσέγγιση του μεγίστου εμβαδού αλλά και του ελαχίστου εμβαδού.

Για 50 σημεία.





Τελος μπορουμε να δουμε απο τις τιμες του εμβαδου οτι προσπαθει να συγγλισει σε μεγαλο η μικρο εμβαδον.

```

● thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz
64
2.95115e+07
15
● thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz
64
3.15021e+07
15
● thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz
64
3.13092e+07
15
○ thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz

● thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz
64
2.5383e+07
15
● thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz
64
2.38687e+07
15
● thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz
64
2.5383e+07
15
○ thanos@thanos-Precision-3571:~/proj/2/GeometricalOptimiz

```

Οπου η πρωτη τιμη ειναι το πληθος των τριγωνων(ασχετο) η δευτερη το εμβαδον του πολυγωνου και η τριτη το πληθος των ακμων.