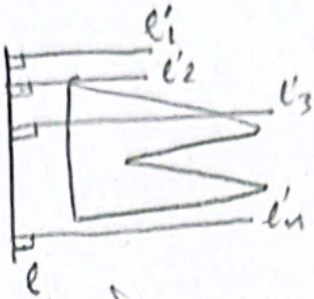
Υπολογιστική Γεωμετρία | Set 1

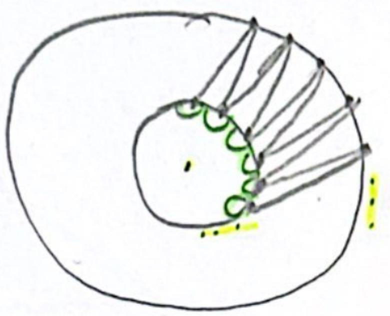
Άσκ 1 (α) Περιγράψτε 1 πολύγωνο P με n κορυφές το οποίο:
- δεν είναι μονότονο ως προς καμία ευθεία του επιπέδου &
- έχει το μικρότερο πλήθος μη-κυρτών κορυφών που μπορείτε να επιτύχετε.

__Λύση:__

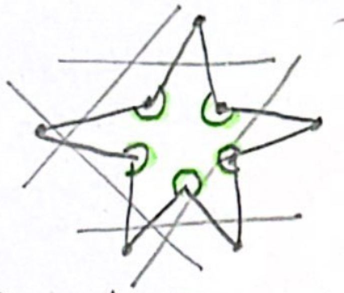


Μονότονα P ως προς την l

- Έστω 2 ομόκεντροι κύκλοι με διαφορετικό r, ο εσωτερικός & ο εξωτερικός
- Χωρίζω τον εσωτερικό κύκλο με n ίσα τόξα & τον εξωτερικό επίσης σε n τόξα.



$\forall$ n (Γενική περίπτωση $n \geq 3$)



Για (5) μη-κυρτές κορυφές ισχύει προφανώς "Αστέρι" που "σχηματίζεται"

Πως μπορεί να γίνει με (2) μη-κυρτές? [Γίνεται από Παλιό]

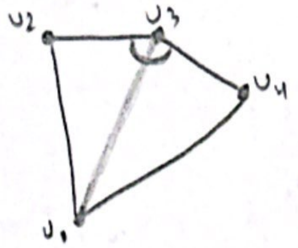
Άσκηση 2 (β) Περιγράψτε 1 P με n κορυφές το οποίο:

- επιδέχεται μοναδική διαίρεση σε τρίγωνα &
- έχει το μικρότερο πλήθος μη κυρτών κορυφών που μπορεί να επιτύχετε & δείξτε τη μοναδική διαίρεση του σε τρίγωνα.

<u>Λύση</u>:
> Γνωρίζουμε ότι τα κυρτά πολύγωνα διαιρούνται σε τρίγωνα εξαιρετικά εύκολα.

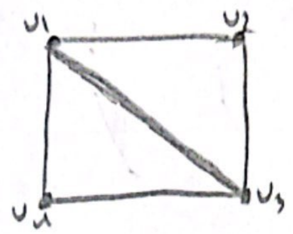
> Επιλέγουμε μια κορυφή u & ύστερα φέρνουμε διαγωνίους από την u προς κάθε άλλη κορυφή εκτός φυσικά από τις γειτονικές της u.

> Αυτή η ευκολία της διαίρεσης ενός κυρτού πολυγώνου σε τρίγωνα είναι & ο λόγος που θέλουμε να χωρίσουμε 1 μη-κυρτό πολύγωνο σε κυρτά τμήματα & ύστερα ακολουθεί η διαγωνιοποίηση.



u_1u_3 η διαγώνιος
∃ 1 μη-κυρτή κορυφή

$n = 4$



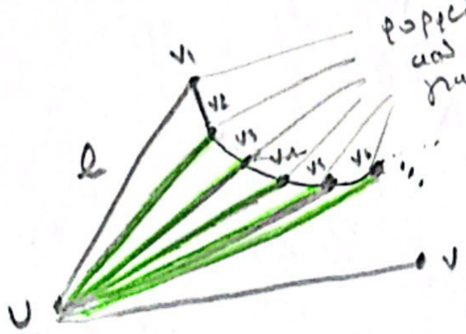
Τετράγωνο
u_1u_3 η διαγώνιος
∄ μη-κυρτές κορυφές

> Στην Γενική περίπτωση: Η u πρέπει να είναι στην απέναντι πλευρά (του συνόρου του P) με τις μη κυρτές κορυφές.

> Σε αυτή την περίπτωση η v θα πρέπει να είναι το χαμηλότερο άκρο της λοξής ακμής e.

popped & pushed από τη στοίβα S για τη διαγωνιοποίηση

(S: κρατάει στοιχεία κορυφές που έχουμε συναντήσει αλλά δεν έχουν συμμετοχή στην τριγωνοποίηση ακόμη.)

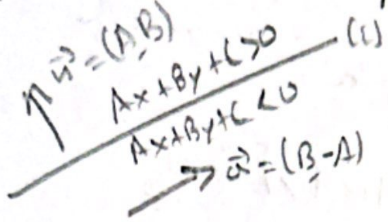


Άσκηση 7) Περιγράψτε σε κώδικα ή ψευδο-κώδικα συνάρτηση **above_line** με ορίσματα double αριθμούς px, py, q1x, q1y, q2x, q2y η οποία, για ευθεία L που ορίζεται από τα σημεία $q_1=(q_{1x}, q_{1y})$ & $q_2=(q_{2x}, q_{2y})$.

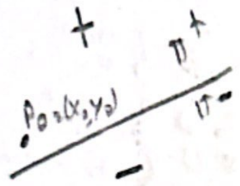
- ▷ Εάν η ευθεία L είναι <u>κατακόρυφη</u>, εκτυπώνει "vertical line" & return -5.
- ▷ Εάν η ευθεία L <u>δεν</u> είναι <u>κατακόρυφη</u> return 1, 0 ή -1 αν το σημείο $p=(p_x, p_y)$ βρίσκεται <u>ψηλότερα από την L</u>, <u>ακριβώς σε αυτήν</u>, ή <u>χαμηλότερα</u> από αυτήν <u>αντίστοιχα</u>.

<u>Λύση</u>   <u>Θεωρία</u>

$$Ax+By+C=0 \quad , (A,B)\neq(0,0)$$



$$\vec{\alpha} \parallel (\varepsilon) \quad , \quad \vec{\alpha} \perp \vec{n}$$

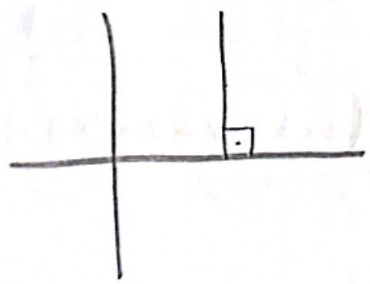


$$P_0 \in \Pi^+ \iff Ax_0+By_0+C>0$$

$$P_0 \in \Pi^- \iff Ax_0+By_0+C<0$$

$$=0 \in (\varepsilon)$$

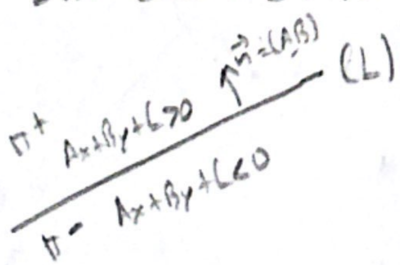
> Έστω ότι τα σημεία $\underline{q_1}=(q_{1x}, q_{1y})$ & $\underline{q_2}=(q_{2x}, q_{2y})$ ορίζουν ευθεία



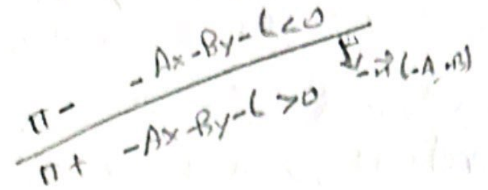
Η l είναι <u>κατακόρυφη</u> $\iff q_{1x}=q_{2x}$ (vertical line)

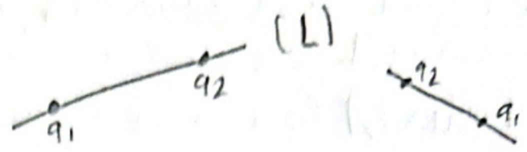
(Όμοια λογική για <u>οριζόντια</u> $\iff q_{1y}=q_{2y}$)

> Έστω ευθεία $L: Ax+By+C=0$ (L)



> Αντίστοιχα $L: \; -Ax - By - C = 0 \quad (2)$





$\underline{q_1} = (q_{1x}, q_{1y})$ & $\underline{q_2} = (q_{2x}, q_{2y})$

$$\begin{array}{ccc} + & - & + \end{array}$$
$$\begin{vmatrix} x & y & 1 \\ q_{1x} & q_{1y} & 1 \\ q_{2x} & q_{2y} & 1 \end{vmatrix} = 0 \iff$$

$$(q_{1y} - q_{2y}) \cdot x - (q_{1x} - q_{2x}) \cdot y + (q_{1x} \cdot q_{2y} - q_{2x} \cdot q_{1y}) = \underline{\underline{0}}$$

$P = (P_x, P_y) \implies$

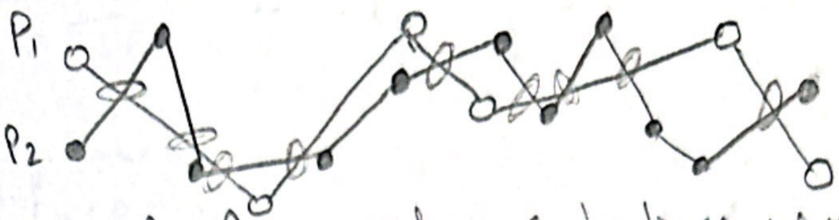
$$(q_{1y} - q_{2y}) \cdot P_x - (q_{1x} - q_{2x}) \cdot P_y + (q_{1x} \cdot q_{2y} - q_{2x} \cdot q_{1y}) \begin{cases} > 0 & \text{από πάνω} \\ < 0 & \text{από κάτω} \\ == 0 & \text{πάνω στην L} \end{cases}$$

```
int above_line(double px, double py, double q1x, double q1y, double q2x, double q2y){
    if(q1x == q2x){
        print("Vertical Line");
        return -5;
    }
    double value = (q1y-q2y)*p.x - (q1x-q2y)*p.y + (q1x*q2y - q2x*q1x);
    if(value >0){
        return 1;
    }else if(value<0){
        return -1;
    }else{
        return 0;
    }
```

Είναι να κάνουμε έλεγχο ως προς <u>περιθώριο ασφαλείας</u>.

Άσκ3 Δίδονται 2 μονότονες πολυγωνικές γραμμές P_1 & P_2 με n κορυφές συνολικά.

(α)

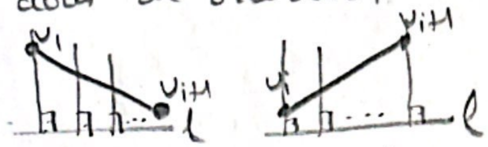


Ν.Δ.Ο. αν οι P_1, P_2 είναι & οι 2 μονότονες ως προς την ίδια (πχ: τον οριζόντιο) διεύθυνση (σχήμα), τότε τέμνονται σε $\boxed{O(n)}$ σημεία.

Λύση:
- Έστω P_1: n_1 κορυφές, P_2: n_2 κορυφές
- $\boxed{n_1 + n_2 = n}$

- Μια πολυγωνική γραμμή είναι μονότονη ως προς 1 διεύθυνση αν το "ανεβοκατέβασμα" από μια κορυφή v_i στην v_{i+1} αυξάνεται ως προς αυτή τη διεύθυνση

Δηλ:



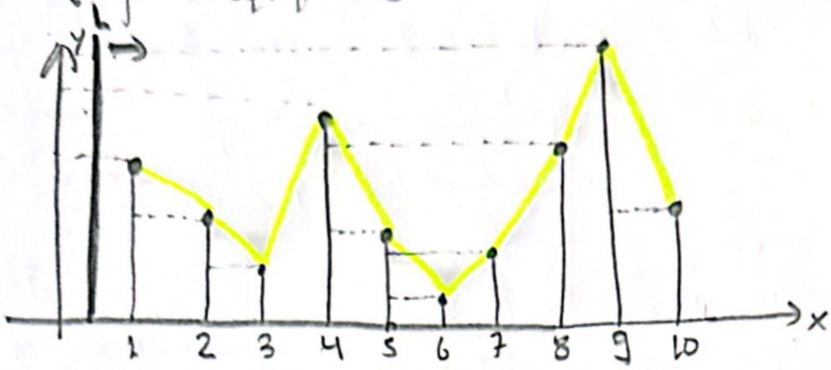
- Θα κάνουμε χρήση της ιδέας της sweep-line.
- Η σάρωση στο δεδομένο πρόβλημα θα "προχωρά" από αριστερά προς τα δεξιά. (και από πάνω προς τα κάτω)
- Δηλ: Αντί να σαρώνουμε από τη y_{max} σε τη y_{min}, θα σαρώνουμε από τη x_{min} προς τη x_{max}.

- Συνεπώς: Κατά τη διάρκεια της σάρωσης κάθε φορά που συναντάμε 1 κορυφή (της P_1 ή της P_2) ελέγχουμε αν αυτή η κορυφή είναι το αριστερό άκρο (endpoint) ακμής που φθίνει ή αυξάνει. Μπορούμε να κρατάμε αυτή την πληροφορία για κάθε ακμή.

Άσκηση 4) Δίνεται σύνολο S κατακόρυφων ευθ. τμημάτων για τα οποία ισχύει ότι το 1 άκρο τους βρίσκεται στον θετικό ημιάξονα των x & το άλλο άκρο τους έχει θετική y-συντεταγμένη (σχήμα).

Θέλουμε να καθορίσουμε για ∀ ευθ. τμήμα $s \in S$, τον αύξοντα αριθμό του ευθ. τμήματος το οποίο βλέπει ένας παρατηρητής τοποθετημένος στο άνω άκρο του s ο οποίος κοιτάζει οριζόντια & προς τα αριστερά.

(Αν ο παρατηρητής δεν βλέπει κάποιο ευθ. τμήμα, τότε ο μηδενικός αύξοντας αριθμός = 0)



Έξοδος:

0	1	2	0	4	5	5	4	0	9
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
1	2	3	4	5	6	7	8	9	10

Εφαρμόστε τη μέθοδο σάρωσης επιπέδου για να περιγράψετε αλγόριθμο $O(|S|)$

- Η είσοδος είναι η ακολουθία των συντεταγμένων των άνω άκρων των ευθ. τμημάτων με τη σειρά από αριστερά προς τα δεξιά.
- Η έξοδος είναι η ακολουθία των αυξόντων αριθμών των ευθ. τμημάτων

Λύση:
- Συνδέοντας με ακμές τις κορυφές παίρνουμε μια ανοικτή πολυγωνική γραμμή
- Σημεία ενδιαφέροντος για εμάς είναι οι κορυφές
- Καθώς το L "περνάει" από αριστερά προς τα δεξιά θα βρίσκει τις κορυφές v_i και για ∀ v_i θα αποθηκεύει σε 1 λίστα το (x_i, y_i).
- Στο τέλος της "σάρωσης" θα έχουμε 1 λίστα

(x_1,y_1)	(x_2,y_2)	...	(x_n,y_n)
1	2		n

index

(Κόστος $O(|S|)$ για S κορυφές)

- Για την κορυφή 1: print("1: 0") // Δεν θα χρειαστεί έλεγχος
- Για την κορυφή 2: if($y_1 > y_2$) => list2{1}, sum(list2)=1, print("2: 1")
- Για την κορυφή n:

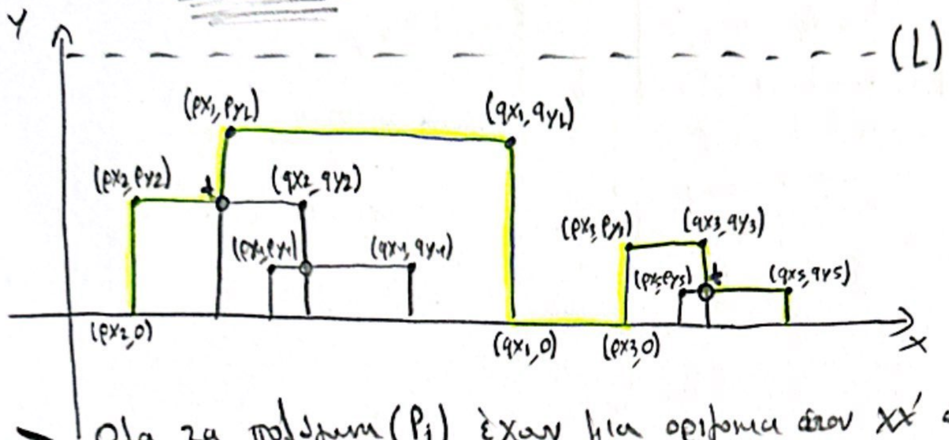
```
sum=0
for (int i=n-1, i=1, i--){
    if (yn < yi){
      sum ++;   // Αυτό 1 αριθμό θα δίνει, όχι λίστα κορυφών
    }
}
print("n: sum");
```

$\boxed{O(n^2)}$

- Δεν εφαρμόζεται ακριβώς σάρωση επιπέδου.
- Στη σάρωση επιπέδου αποθηκεύονται πληροφορίες για τα αντικείμενα που έχουν σχεδιαστεί σε κατάλληλη δομή & ύστερα την χρησιμοποιούμε για την επεξεργασία του τρέχοντος αντικειμένου

Ασκ5 Η skyline μιας πόλης ορίζεται ως το περίγραμμα των κτιρίων της πόλης, όπου θεωρούμε ότι η πόλη περιγράφεται από n ορθογώνια παραλληλόγραμμα (= τα κτίρια) καθένα από τα οποία έχει τη μια πλευρά του επάνω στον άξονα x.
Για δοθέν σύνολο κτιρίων, θέλουμε να εκτυπώσουμε τις συντεταγμένες των κορυφών της skyline αυτών των κτιρίων κατά σειράν από αριστερά προς τα δεξιά.

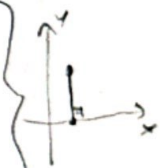
> Χρησιμοποιώντας την ορθόγωνια γραμμή σάρωσης, θα σαρώσουμε τα τετράγωνα πολύγωνα (ορθογώνια)
> Ταξινόμηση (Sorting) των κορυφών (κατά φθίνουσα y-συντεταγμένη).
⟹ $O(n\log n)$ χρόνο (n το # των κορυφών)



> Όλα τα πολύγωνα (P_i) έχουν μια οριζόντια στον xx' ακμή.
> Για ∀ πολύγωνο αποθηκεύω τα points: (px_i, py_i) & (qx_i, qy_i)
> Έχουμε διάφορες περιπτώσεις:

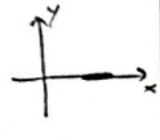
case 1 > Αν (px_i, py_i) αριστερά από ∀ άλλη κορυφή:
Ζωγραφίζουμε το κατακόρυφο ευθ. τμήμα με κορυφές: (px_i, qy_i) & $(px_i, 0)$

case 2 > Αν (px_i, py_i) δεξιότερα από ∀ άλλη κορυφή:
Ζωγραφίζουμε το κατακόρυφο ευθ. τμήμα με κορυφές: (qx_i, qy_i) & $(qx_i, 0)$



case 3 > Αν για το P πολύγωνο με κορυφές (px_i, py_i) & (qx_i, qy_i).
Ισχύει ότι βρίσκεται "μέσα" σε ένα μεγαλύτερο πολύγωνο το αγνοούμε.
Δηλ.: $(px_j < px_i \ \&\& \ qx_j > qx_i)$ && $(py_j > py_i \ \&\& \ qy_j > qy_i)$

case 4 > Αν ∃ ένα συνεχές διάστημα $[x_a, x_b]$ στο οποίο ∄ ορθογώνιο, τότε ζωγραφίζουμε το ευθ. τμήμα $[(x_a, 0), (x_b, 0)]$



case 5 > Αν ∃ σημείο τομής μεταξύ 2 πολυγώνων, ζωγραφίζουμε το ευθ. τμήμα:
$[(px_i, py_i), (tx_i, ty_i)]$ ή $[(tx_i, ty_i), (qx_i, qy_i)]$

Άσκ 6 Ν.Δ.Ο. ένα πολύγωνο P μπορεί να τριγωνοποιηθεί

> Έστω πολύγωνο P με h τρύπες, n κορυφές συνολικά.
> Όταν λέμε συνολικά: Κορυφές των τρυπών + Κορυφές του πολυγώνου.
> Τότε μια τριγωνοποίηση του P έχει $\boxed{2h + n - 2}$ τρίγωνα.

> Πως:

> Έχουμε $|V| = n$ κορυφές

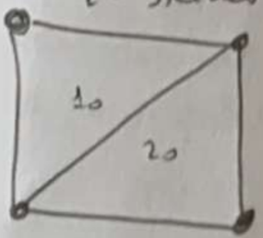
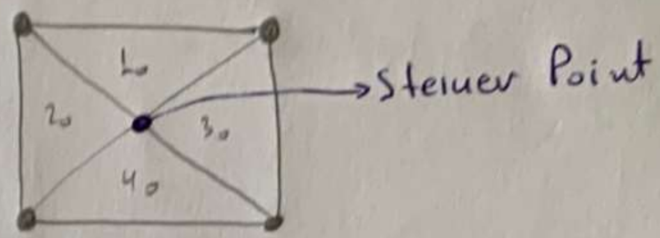
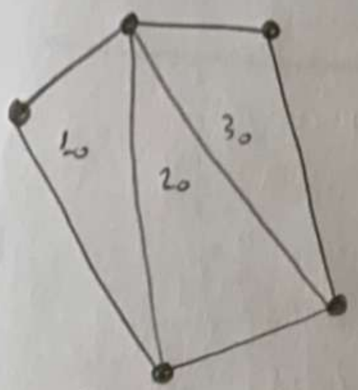
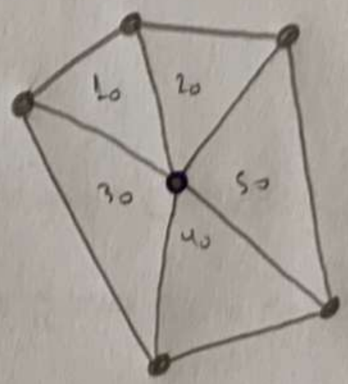
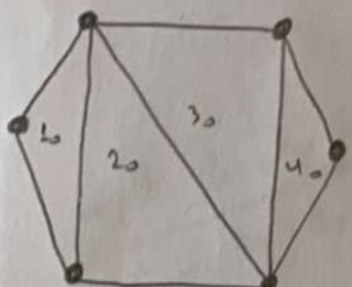
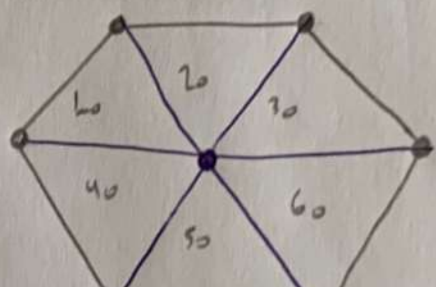
> Έχουμε $\underbrace{h}_{\text{holes}} + \underbrace{t}_{\text{triangles}} + 1$ faces, ένα για $\forall$ τρίγωνο & τρύπα + το "εξωτερικό" face.

$E = (3t + n)/2$ ακμές, όπου 3 ανά τρίγωνο + το "εξωτερικό" (boundary) "μοιράζει" $\forall$ ακμή επί 2.

$$\underbrace{V}_{\text{vertices}} - \underbrace{E}_{\text{edges}} + \underbrace{F}_{\text{faces}} = 2 \Longrightarrow \boxed{E = n + 2h - 2}$$

Άσκηση 1 (i) > Χωρίς Steiner Points ο γνωστός τρόπος να "σπάσουμε" ένα πολύγωνο σε τρίγωνα είναι με την χρήση διαγωνίων.

> Θα επιχειρήσουμε κατασκευαστική απόδειξη:

Χωρίς Steiner Point(s)	Με Steiner Point
 n-2 τρίγωνα	 n τρίγωνα
 n-2 τρίγωνα	 n τρίγωνα
 n-2 τρίγωνα	 n τρίγωνα

[∃ Το γνωστό θεώρημα που μας εξασφαλίζει τα n-2 τρίγωνα]

[Φαίνεται ότι υπάρχει σχέση και με περισσότερα σημεία Steiner θα "αυξάνει" το πλήθος των τριγώνων]