

Camp Προετοιμασίας 23ου ΠΔΠ

Υπολογιστική Γεωμετρία

Γιάννης Χατζημύχος
(feedward@gmail.com)

14 Απριλίου 2011

Υπολογιστική Γεωμετρία

Προβλήματα:

- Τομές ευθύγραμμων τμημάτων, σχετικές θέσεις σημείων κτλ.
- Κυρτό περίβλημα
- Αναζήτηση ανά περιοχή (range searching)

Προαπαιτούμενες γνώσεις:

- Διανύσματα
- Ορίζουσες (3×3)

Ορίζουσες

- Υπολογισμός ορίζουσας 2x2:

$$\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} =$$

$$x_1 y_2 - y_1 x_2$$

- <Γινόμενο πρώτης διαγωνίου> - <Γινόμενο δεύτερης διαγωνίου>

Ορίζουσες

- Υπολογισμός ορίζουσας 3x3:

$$\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}$$

- Ανάπτυξη κατά τα στοιχεία της πρώτης γραμμής

Ορίζουσες

1^ο Βήμα

$$\begin{vmatrix} \textcircled{x_1} & \cancel{y_1} & \cancel{z_1} \\ \cancel{x_2} & y_2 & z_2 \\ \cancel{x_3} & y_3 & z_3 \end{vmatrix}$$



$$+ x_1 \begin{vmatrix} y_2 & z_2 \\ y_3 & z_3 \end{vmatrix}$$

2^ο Βήμα

$$\begin{vmatrix} \cancel{x_1} & \textcircled{y_1} & \cancel{z_1} \\ x_2 & \cancel{y_2} & z_2 \\ x_3 & \cancel{y_3} & z_3 \end{vmatrix}$$



$$- y_1 \begin{vmatrix} x_2 & z_2 \\ x_3 & z_3 \end{vmatrix}$$

3^ο Βήμα

$$\begin{vmatrix} \cancel{x_1} & \cancel{y_1} & \textcircled{z_1} \\ x_2 & y_2 & \cancel{z_2} \\ x_3 & y_3 & \cancel{z_3} \end{vmatrix}$$



$$+ z_1 \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix}$$

Ορίζουσες

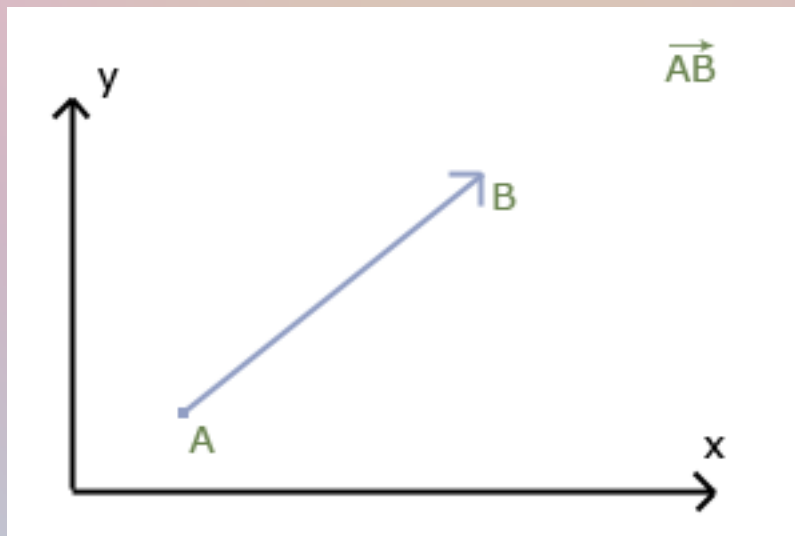
$$\begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} =$$

$$+ x_1 \begin{vmatrix} y_2 & z_2 \\ y_3 & z_3 \end{vmatrix} - y_1 \begin{vmatrix} x_2 & z_2 \\ x_3 & z_3 \end{vmatrix} + z_1 \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} =$$

$$x_1(y_2 z_3 - z_2 y_3) - y_1(x_2 z_3 - z_2 x_3) + z_1(x_2 y_3 - y_2 x_3)$$

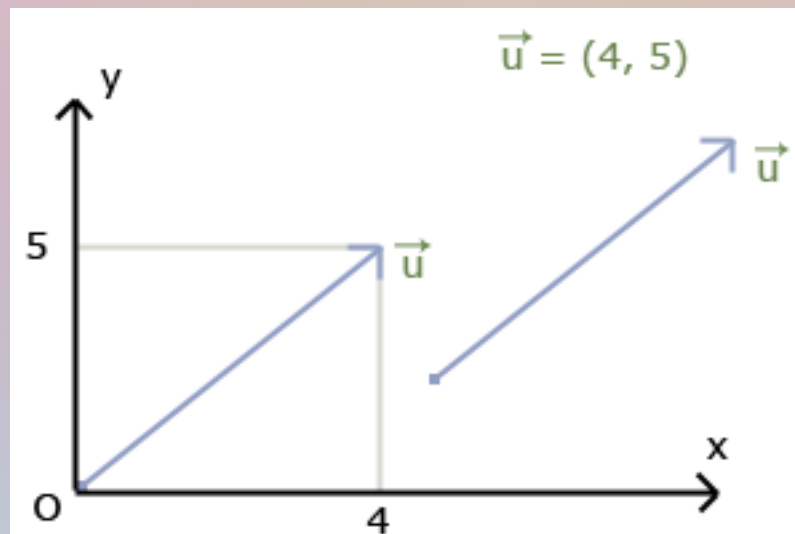
Διανύσματα

- Διάνυσμα είναι το γεωμετρικό αντικείμενο που έχει **μέτρο** και **κατεύθυνση**.
- Σχεδιάζεται σαν ένα προσανατολισμένο ευθύγραμμο τμήμα.



Διανύσματα

- **Ελεύθερα** διανύσματα: Δεν έχουν σημείο εφαρμογής
- Μπορούν να συμβολιστούν με τις συντεταγμένες του τερματικού τους σημείου όταν το αρχικό τους σημείο βρίσκεται στο $(0,0)$.



Ιδιότητες διανυσμάτων

Έστω $\mathbf{u} = (x_1, y_1)$, $\mathbf{v} = (x_2, y_2)$

- $\mathbf{u} + \mathbf{v} = (x_1 + x_2, y_1 + y_2)$

- $\mathbf{u} - \mathbf{v} = (x_1 - x_2, y_1 - y_2)$

- $\lambda \mathbf{u} = (\lambda x_1, \lambda y_1)$

- $|\mathbf{u}| = \sqrt{x^2 + y^2}$

- **Εσωτερικό γινόμενο:**

$$\mathbf{u} \cdot \mathbf{v} = x_1 x_2 + y_1 y_2 = |\mathbf{u}| |\mathbf{v}| \cos \theta$$

Ιδιότητες διανυσμάτων

Έστω $\mathbf{u} = (x_1, y_1)$, $\mathbf{v} = (x_2, y_2)$

- **Μοναδιαία διανύσματα:**

$$\mathbf{i} = (1, 0, 0)$$

$$\mathbf{j} = (0, 1, 0)$$

$$\mathbf{k} = (0, 0, 1)$$

$$\text{π.χ. } \mathbf{u} = x_1 \mathbf{i} + y_1 \mathbf{j} = x_1(1, 0) + y_1(0, 1)$$

$$= (x_1, 0) + (0, y_1) = (x_1, y_1)$$

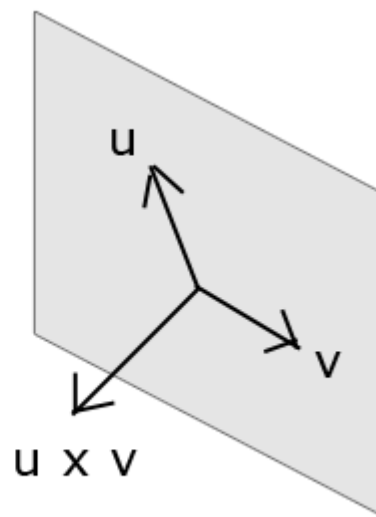
Ιδιότητες διανυσμάτων

Έστω $\mathbf{u} = (x_1, y_1, z_1)$, $\mathbf{v} = (x_2, y_2, z_2)$

- Εξωτερικό γινόμενο:

$$|\mathbf{u} \times \mathbf{v}| = |\mathbf{u}||\mathbf{v}|\sin\theta$$

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix}$$



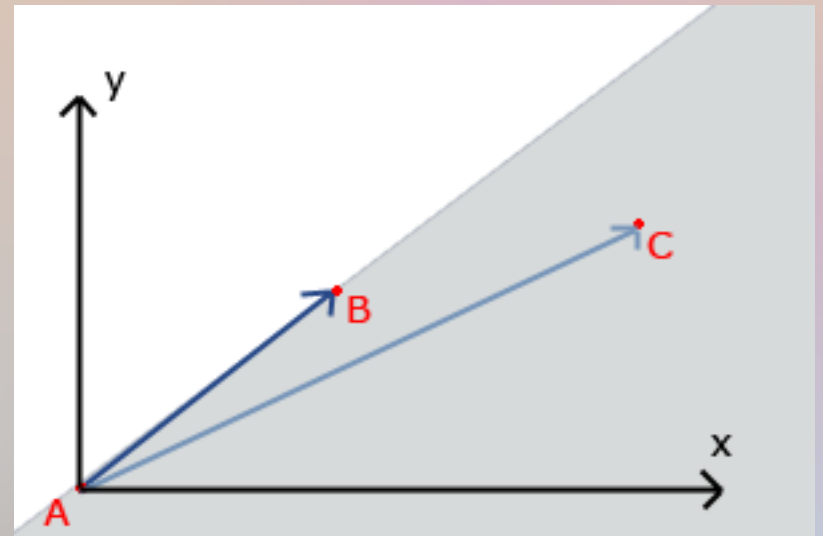
Προσανατολισμός σημείων

Έστω σημεία: $A(x_0, y_0)$, $B(x_1, y_1)$, $C(x_2, y_2)$

- Σημείο C ως προς ευθεία AB:
 - Υπολογίζουμε το εξωτερικό γινόμενο:

$$\mathbf{AB} \times \mathbf{AC}$$

- Ελέγχουμε το πρόσημο της z συνιστώσας:
 - $z > 0$: αριστερά
 - $z < 0$: δεξιά



Προσανατολισμός σημείων

$$\mathbf{AB} = \mathbf{OB} - \mathbf{OA} = (x_1 - x_0, y_1 - y_0)$$

$$\mathbf{AC} = \mathbf{OC} - \mathbf{OA} = (x_2 - x_0, y_2 - y_0)$$

$$\mathbf{AB} \times \mathbf{AC} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_1 - x_0 & y_1 - y_0 & 0 \\ x_2 - x_0 & y_2 - y_0 & 0 \end{vmatrix} =$$

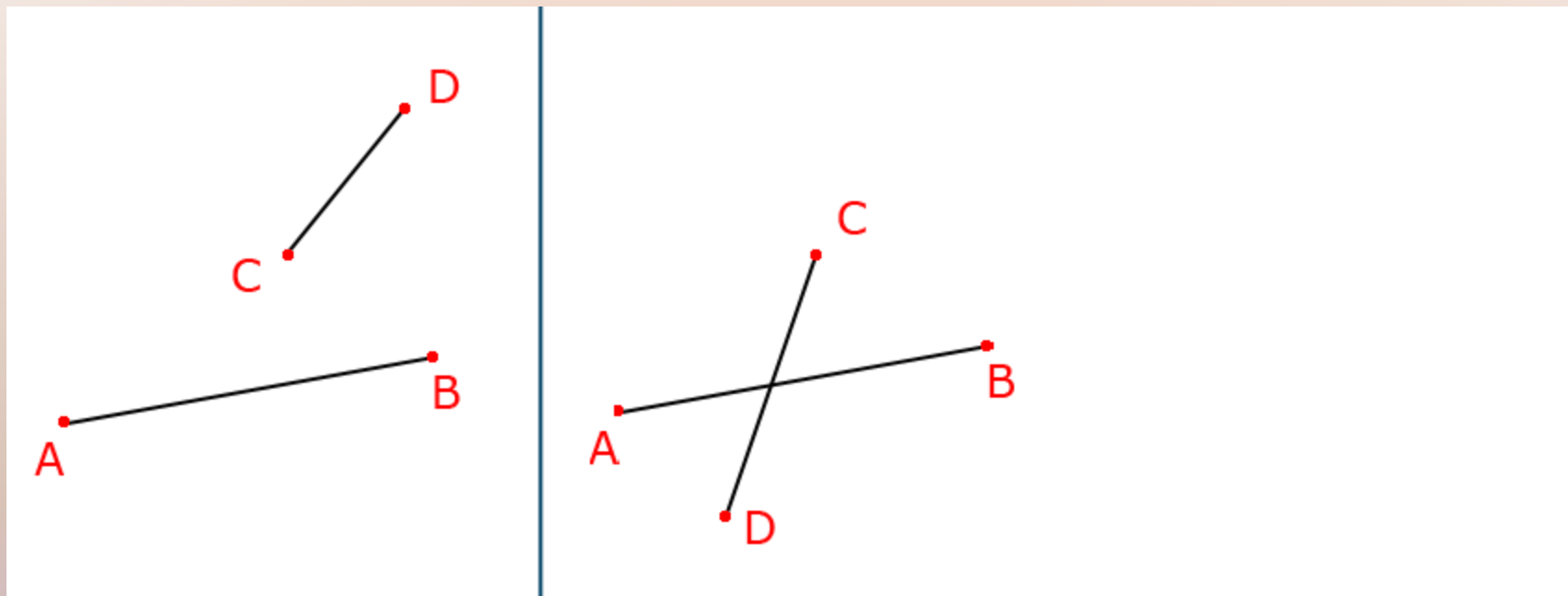
$$[(x_1 - x_0)(y_2 - y_0) - (y_1 - y_0)(x_2 - x_0)] \mathbf{k}$$

Προσανατολισμός σημείων

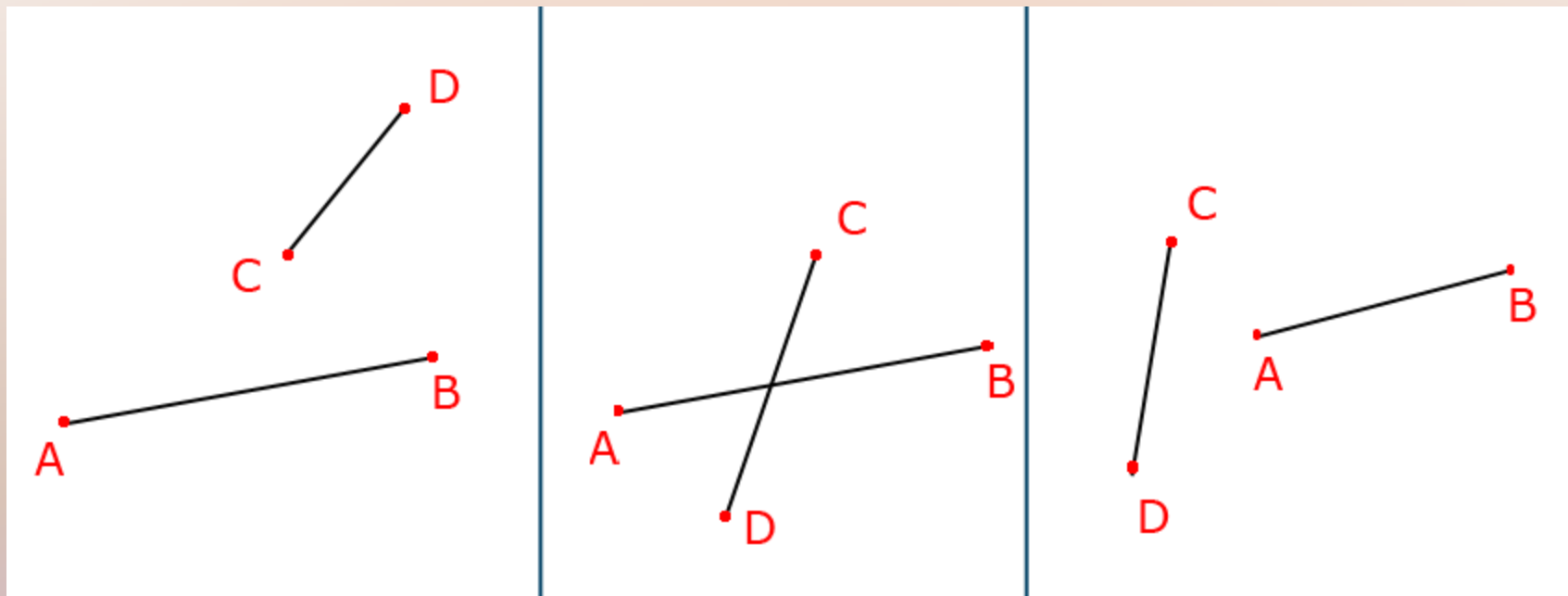
```
//προσανατολισμός του σημείου C ως προς την ευθεία AB  
  
typedef struct point {  
    double x;  
    double y;  
} point;  
  
double CCW(point A, point B, point C) {  
    return (B.x-A.x) * (C.y-A.y) - (B.y-A.y) * (C.x-A.x);  
}
```

Τύπος Συντεταγμένων	Τύπος CCW
double	double
int	long long int

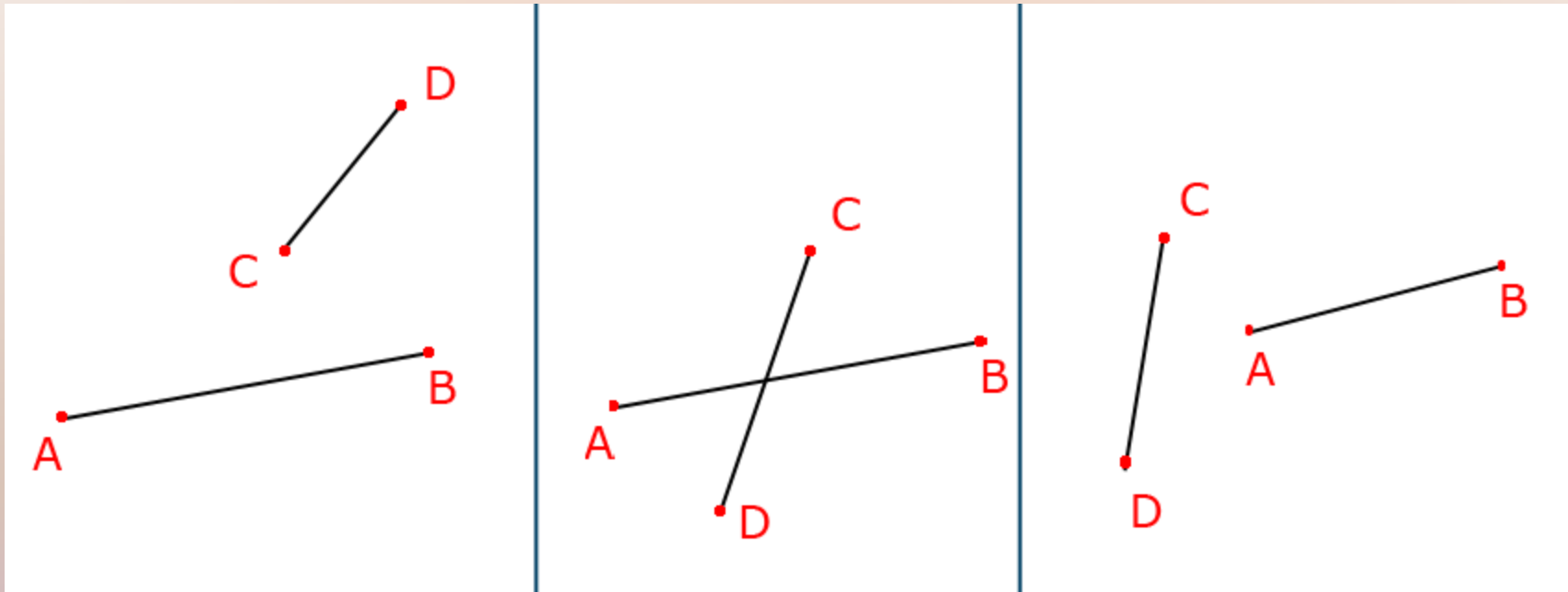
Τομή ευθύγραμμων τμημάτων



Τομή ευθύγραμμων τμημάτων



Τομή ευθύγραμμων τμημάτων



// έλεγχος τομής ευθ. Τμημάτων AB και CD (επιστρέφει 1 ή 0)

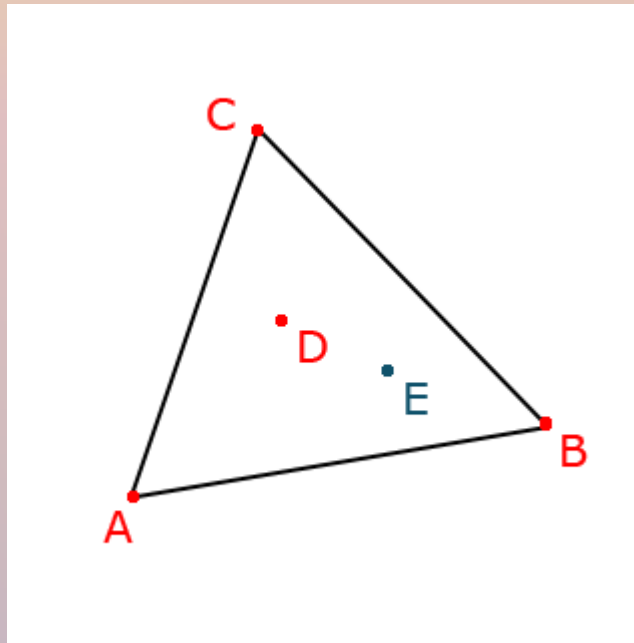
```
int intersect(point A, point B, point C, point D) {  
    return ( CCW(A,B,C) * CCW(A,B,D) < 0 &&  
            CCW(C,D,A) * CCW(C,D,B) < 0 );  
}
```

Σημείο ως προς τρίγωνο

- Δίνεται τρίγωνο ABC και σημείο D . Βρίσκεται το D μέσα στο τρίγωνο;

Σημείο ως προς τρίγωνο

- Δίνεται τρίγωνο ABC και σημείο D . Βρίσκεται το D μέσα στο τρίγωνο;



- Το E βρίσκεται σίγουρα μέσα στο τρίγωνο.

Σημείο ως προς τρίγωνο

Σημείο που βρίσκεται σίγουρα μέσα στο τρίγωνο

Βαρύκεντρο: $E([x_1+x_2+x_3]/3 , [y_1+y_2+y_3]/3)$

1η Λύση

// βρίσκεται το D μέσα στο τρίγωνο ABC; (επιστρέφει 1 ή 0)

```
int inTriangle(point A, point B, point C, point D) {  
    double E;
```

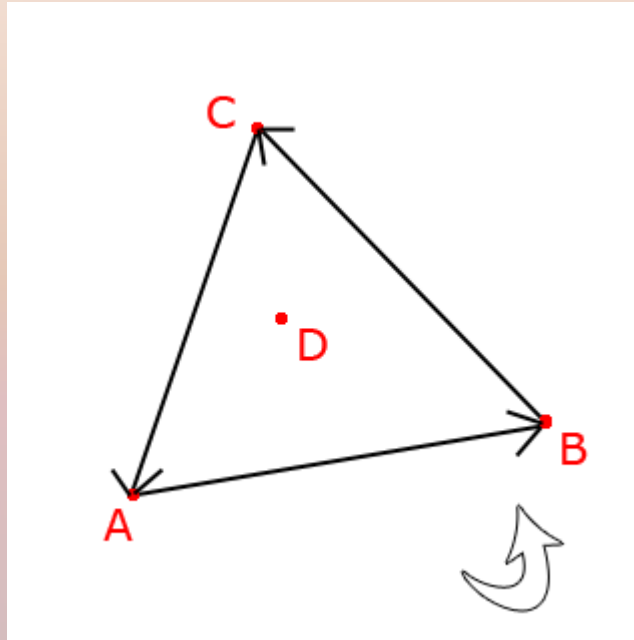
```
    E.x = (A.x + B.x + C.x) / 3;
```

```
    E.y = (A.y + B.y + C.y) / 3;
```

```
    return ( CCW(A,B,D) * CCW(A,B,E) > 0 &&  
            CCW(B,C,D) * CCW(B,C,E) > 0 &&  
            CCW(A,C,D) * CCW(A,C,E) > 0 );
```

```
}
```

Σημείο ως προς τρίγωνο



2η Λύση

// βρίσκεται το D μέσα στο τρίγωνο ABC; (επιστρέφει 1 ή 0)

```
int inTriangle(point A, point B, point C, point D) {  
    return ( CCW(A,B,D) * CCW(B,C,D) > 0 &&  
            CCW(B,C,D) * CCW(C,A,D) > 0 );  
}
```

Σημείο ως προς κυρτό πολύγωνο

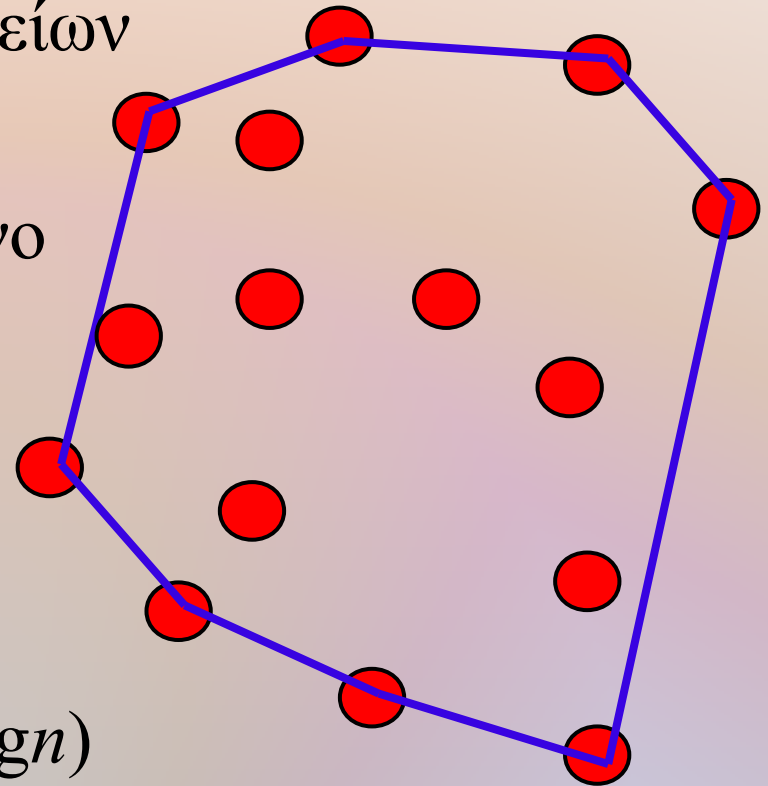
- Και οι δύο προηγούμενες λύσεις γενικεύονται στα κυρτά πολύγωνα.
- 1η Λύση:
 - Βρίσκουμε ένα σημείο E που είναι σίγουρα μέσα στο πολύγωνο.
 - Ελέγχουμε αν τα σημεία D και E βρίσκονται στο ίδιο ημιεπίπεδο ως προς κάθε πλευρά.
- 2η Λύση:
 - Ορίζουμε φορά και διασχίζουμε το πολύγωνο, ελέγχοντας αν το D μένει στο ίδιο ημιεπίπεδο.

Άλλα προβλήματα

- Είναι τρία σημεία A, B, C συνευθειακά;
 - Ελέγχουμε αν τα διανύσματα \mathbf{AB}, \mathbf{AC} είναι παράλληλα: $\mathbf{AB} \times \mathbf{AC} = 0$
- Είναι δύο ευθύγραμμα τμήματα AB και CD κάθετα μεταξύ τους;
 - Ελέγχουμε αν τα διανύσματα \mathbf{AB}, \mathbf{CD} είναι κάθετα: $\mathbf{AB} \cdot \mathbf{CD} = 0$
- Είναι δύο ευθύγραμμα τμήματα AB και CD παράλληλα μεταξύ τους;
 - Ελέγχουμε αν τα διανύσματα \mathbf{AB}, \mathbf{CD} είναι παράλληλα: $\mathbf{AB} \times \mathbf{CD} = 0$

Κυρτό περίβλημα

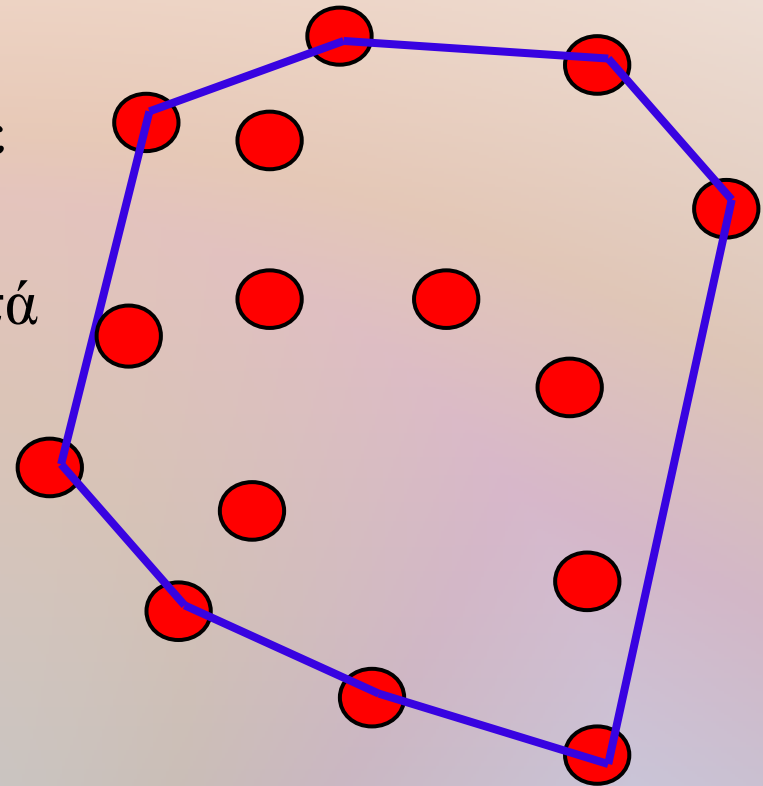
- Είσοδος:
 - Σύνολο $S = \{s_1, \dots, s_n\}$ n σημείων
- Έξοδος:
 - Το μικρότερο κυρτό πολύγωνο που περιέχει όλα τα σημεία
- Πολλοί αλγόριθμοι
 - Naïve – $O(n^3)$
 - Insertion – $O(n \log n)$
 - Divide and Conquer – $O(n \log n)$
 - Gift Wrapping – $O(nh)$, h = no of points on the hull
 - **Graham Scan – $O(n \log n)$**



Graham Scan

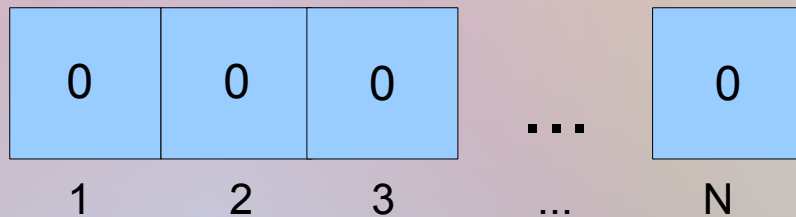
Βήματα

- Βρίσκουμε το χαμηλότερο σημείο (με την μικρότερη y συντεταγμένη)
- Ταξινομούμε τα υπόλοιπα σημεία κατά γωνία γύρω από αυτό το σημείο
- Σκανάρουμε τα σημεία με αντιωρολογιακή φορά (CCW) order
 - Πετούμε όσα σημεία κάνουν ωρολογιακή (CW) στροφή



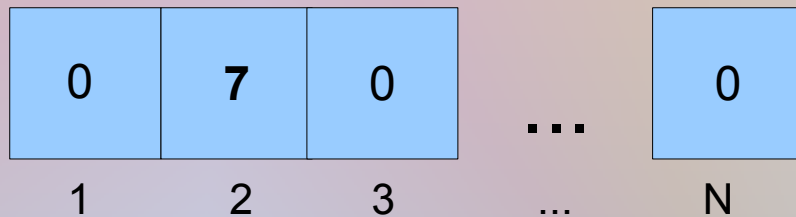
Range Searching

- Δίνεται ένας πίνακας N ακεραίων (αρχικά όλοι 0). Υλοποιήστε τις παρακάτω πράξεις:
 - **ADD X NUM**: πρόσθεσε τον αριθμό NUM στην θέση X).
 - **SUM X Y**: επέστρεψε το άθροισμα των αριθμών στο διάστημα $X...Y$



Range Searching

- Δίνεται ένας πίνακας N ακεραίων (αρχικά όλοι 0). Υλοποιήστε τις παρακάτω πράξεις:
 - **ADD X NUM**: πρόσθεσε τον αριθμό NUM στην θέση X).
 - **SUM X Y**: επέστρεψε το άθροισμα των αριθμών στο διάστημα $X...Y$



ADD 2 7

Range Searching

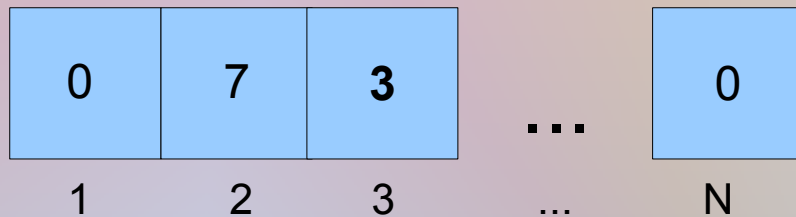
- Δίνεται ένας πίνακας N ακεραίων (αρχικά όλοι 0). Υλοποιήστε τις παρακάτω πράξεις:
 - **ADD X NUM**: πρόσθεσε τον αριθμό NUM στην θέση X).
 - **SUM X Y**: επέστρεψε το άθροισμα των αριθμών στο διάστημα $X...Y$



ADD 2 7
ADD 3 1

Range Searching

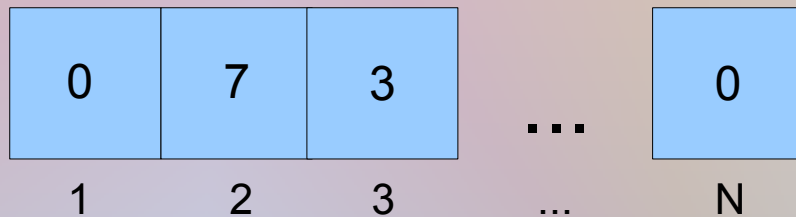
- Δίνεται ένας πίνακας N ακεραίων (αρχικά όλοι 0). Υλοποιήστε τις παρακάτω πράξεις:
 - **ADD X NUM**: πρόσθεσε τον αριθμό NUM στην θέση X).
 - **SUM X Y**: επέστρεψε το άθροισμα των αριθμών στο διάστημα $X...Y$



ADD 2 7
ADD 3 1
ADD 3 2

Range Searching

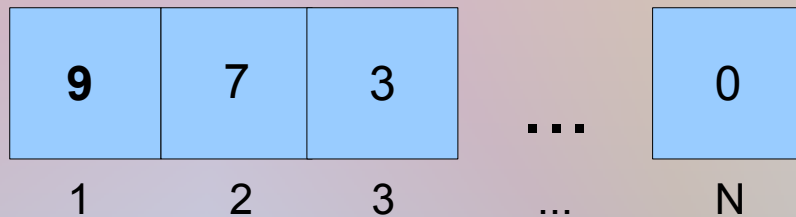
- Δίνεται ένας πίνακας N ακεραίων (αρχικά όλοι 0). Υλοποιήστε τις παρακάτω πράξεις:
 - **ADD X NUM**: πρόσθεσε τον αριθμό NUM στην θέση X).
 - **SUM X Y**: επέστρεψε το άθροισμα των αριθμών στο διάστημα $X...Y$



ADD 2 7
ADD 3 1
ADD 3 2
SUM 1 3 >> 10

Range Searching

- Δίνεται ένας πίνακας N ακεραίων (αρχικά όλοι 0). Υλοποιήστε τις παρακάτω πράξεις:
 - **ADD X NUM**: πρόσθεσε τον αριθμό NUM στην θέση X).
 - **SUM X Y**: επέστρεψε το άθροισμα των αριθμών στο διάστημα $X...Y$

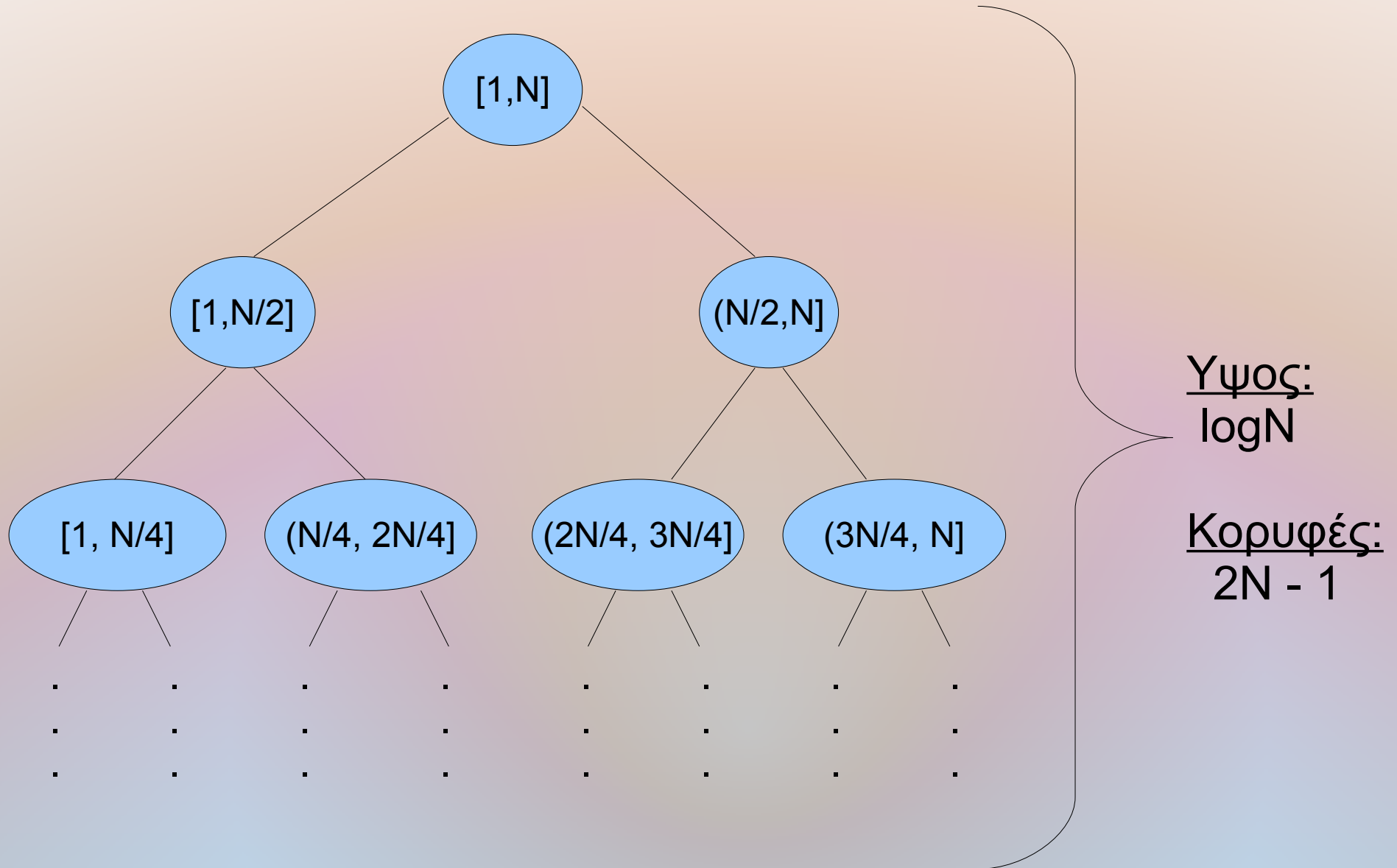


ADD 2 7
ADD 3 1
ADD 3 2
SUM 1 3 >> 10
ADD 1 9

Range Searching

- Προφανής αλγόριθμος:
 - ADD: $O(1)$
 - SUM: $O(N)$
- Μερικά αθροίσματα:
 - ADD: $O(N)$
 - SUM: $O(1)$
- **Bucketing**
- **Interval trees**

Δένδρα Διαστημάτων (interval trees)



Range Searching

- Κατασκευή δέντρου [$O(N)$]:
 - Σταθερός χρόνος για την κατασκευή κάθε κορυφής
- ADD [$O(\log N)$]:
 - Ακολουθούμε το μονοπάτι από το $[1, N]$ στο $[X, X]$ και ανανεώνουμε την κάθε κορυφή σε σταθερό χρόνο
- SUM [$O(\log N)$]:
 - Το αρχικό διάστημα θα σπάσει σε δύο κορυφές το πολύ μία φορά.

Range Searching

- **Bucketing**

- ADD: $O(1)$
- SUM: $O(\sqrt{n})$

- **Interval trees**

- ADD: $O(\log N)$
- SUM: $O(\log N)$
- Preprocessing: $O(N)$

Θέματα υλοποίησης

- Ποτέ σύγκριση μεταβλητών κινητής υποδιαστολής με `==` , `<=` , `>=` , `!=`

ΛΑΘΟΣ

```
double a;  
double b;  
  
if ( a == b ) {  
    //they are equal  
}
```

ΣΩΣΤΟ

```
double a;  
double b;  
  
if (fabs(a-b) < 1e-7) {  
    //they are equal  
}
```

- Προτιμήστε double από float.
- Προσοχή στις ακραίες περιπτώσεις!