

# ΑΛΓΟΡΙΘΜΟΙ 2021-2022

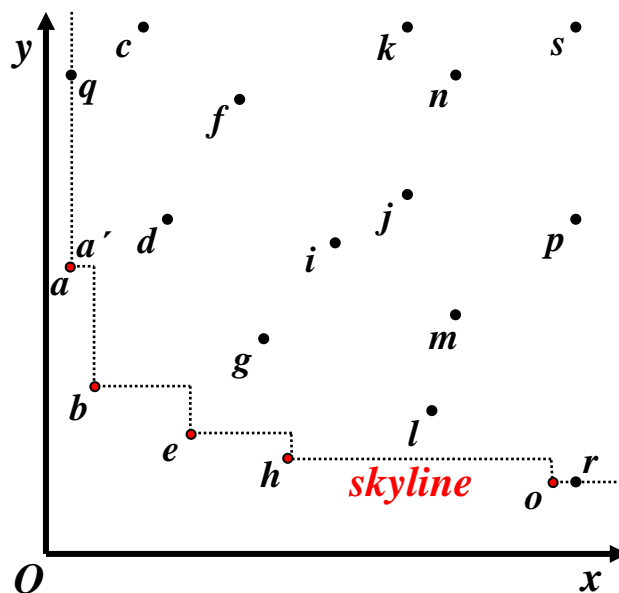
## Α ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ (1 βαθμός)

### Αποδοτικός Αλγόριθμος για ερωτήματα κορυφογραμμής (skyline queries)

**Ορισμός:** Όταν δίνεται μία σχέση κυριαρχίας σε ένα σύνολο δεδομένων, τότε το **ερώτημα κορυφογραμμής επιστρέφει όλα τα αντικείμενα τα οποία δεν κυριαρχούνται από κανένα άλλο αντικείμενο του συνόλου**. Σε περίπτωση που το σύνολο δεδομένων αποτελείται από πολυδιάστατα αντικείμενα, ένα αντικείμενο θα κυριαρχεί πάνω σε ένα άλλο αν είναι το ίδιο ή καλύτερο σε όλες τις διαστάσεις και τουλάχιστον σε μία αυστηρά καλύτερο.

Τυπικά, για δύο δισδιάστατα αντικείμενα  $A(x_1, y_1)$  και  $B(x_2, y_2)$ , θεωρώντας ότι καλύτερη τιμή είναι η ελάχιστη σε κάθε μία από τις δύο διαστάσεις, θα έχουμε ότι το **A κυριαρχεί πάνω στο B** αν και μόνον αν:  $x_1 \leq x_2$  και  $y_1 \leq y_2$  και  $(x_1 \neq x_2 \text{ ή } y_1 \neq y_2)$ .

**Παράδειγμα:** Ένα απλό παράδειγμα είναι όταν τα αντικείμενα είναι σημεία στο Ευκλείδειο επίπεδο, θεωρώντας ότι η καλύτερη τιμή σε κάθε μία από τις δύο διαστάσεις είναι η ελάχιστη. Έστω τα 20 σημεία του παρακάτω σχήματος:  $a(1,12)$ ,  $a'(1,12)$ ,  $b(2,7)$ ,  $c(4,22)$ ,  $d(5,14)$ ,  $e(6,5)$ ,  $f(8,19)$ ,  $g(9,9)$ ,  $h(10,4)$ ,  $i(12,13)$ ,  $j(15,15)$ ,  $k(15,22)$ ,  $l(16,6)$ ,  $m(17,10)$ ,  $n(17,20)$ ,  $o(21,3)$ ,  $p(22,14)$ ,  $q(1,20)$ ,  $r(22,3)$ ,  $s(22,22)$ . Τότε το ερώτημα κορυφογραμμής επιστρέφει το σύνολο των σημείων:  $\{(1,12), (2,7), (6,5), (10,4), (21,3)\}$ .



### Τι πρέπει να υλοποιηθεί

Στην εργασία αυτή πρέπει να υλοποιήσετε έναν αποδοτικό αλγόριθμο για την εύρεση της κορυφογραμμής (skyline) σε σύνολα σημείων δύο διαστάσεων στο Ευκλείδειο επίπεδο.

Ως είσοδος, θα δίνεται ένα αρχείο στην παρακάτω μορφή:

```
20
1 12
1 12
2 7
4 22
5 14
6 5
8 19
9 9
10 4
12 13
15 15
15 22
16 6
17 10
17 20
21 3
22 14
1 20
22 3
22 22
```

Δηλαδή θα περιέχει τα δεδομένα των σημείων όπου στην πρώτη γραμμή θα αναγράφεται το πλήθος  $N$  των σημείων και σε κάθε επόμενη γραμμή θα ακολουθούν οι συντεταγμένες των σημείων, οι οποίες θα είναι ακέραιοι αριθμοί στο διάστημα  $[0...1000]$  χωρισμένοι μεταξύ τους με ένα κενό.

Το πρόγραμμα θα πρέπει να υπολογίζει και να εκτυπώνει στην οθόνη τις συντεταγμένες των σημείων της κορυφογραμμής αυστηρά στην εξής μορφή εξόδου:

```
1 12
2 7
6 5
10 4
21 3
```

(δηλαδή κατά αύξουσα σειρά ως προς την τετμημένη τους).

### Τρόπος υλοποίησης:

- 1) Να χρησιμοποιήσετε αποκλειστικά την γλώσσα **Java**, ακολουθώντας τις οδηγίες για την υποβολή ασκήσεων java στο σύστημα Eagle ([Οδηγίες-Eagle-2022.pdf](#)).
- 2) Το **μοναδικό όρισμα εισόδου** θα είναι το αρχείο εισόδου. Το όρισμα αυτό θα πρέπει να είναι το **args[0]** από τα ορίσματα της command line, το οποίο διαβάζεται από την **main**.
- 3) Για τη λύση να χρησιμοποιήσετε αλγόριθμο που να είναι αποδοτικός [ **$O(N \log N)$** ], με μεθόδους που διδαχθήκατε στο μάθημα (όπως π.χ. **divide&conquer**, κ.α.) ή από τη βιβλιογραφία, καθώς το  $N$  μπορεί να είναι μεγάλο και **θα υπάρχει χρονικός περιορισμός στην εκτέλεση των test cases**.
- 4) Ο πηγαίος κώδικας να έχει **συνοπτικά σχόλια** μέσα στον κώδικα (**inline**) και **σχόλια** επάνω από κάθε συνάρτηση, τα οποία να εξηγούν το σκεπτικό της υλοποίησής σας.

### Παραδοτέο:

- Όλο το πρόγραμμα (πηγαίος κώδικας) θα πρέπει να είναι υλοποιημένο σε **ένα και μοναδικό αρχείο java** το οποίο θα έχει όνομα **Skyline.java** (κλάση Skyline).
- Το αρχείο αυτό θα πρέπει να το ανεβάσετε και να το **υποβάλετε** στο σύστημα **Eagle** (eagle.csd.auth.gr) με τον λογαριασμό σας.
- Στο σύστημα Eagle θα γίνεται **αυτόματη εκτέλεση** του κώδικα σας σε διάφορα test cases (και φανερά αλλά και κρυφά) καθώς και **αυτόματη βαθμολόγηση** (μέγιστος βαθμός το **100**).
- Μπορείτε να υποβάλετε στο σύστημα μέχρι και 20 φορές. Συνεπώς θα πρέπει να έχετε κάνει την υλοποίηση πρώτα σε ένα άλλο περιβάλλον της επιλογής σας (π.χ. Netbeans, IntelliJ, κλπ.) και να το έχετε δοκιμάσει ώστε **να τρέχει σωστά** και **να παράγει σωστά αποτελέσματα** πριν κάνετε υποβολές.
- Έναρξη υποβολών: **Δευτέρα 2/5/2022 10:00πμ.**
- Τέλος υποβολών: **Πέμπτη 12/5/2022 11:59μμ.**

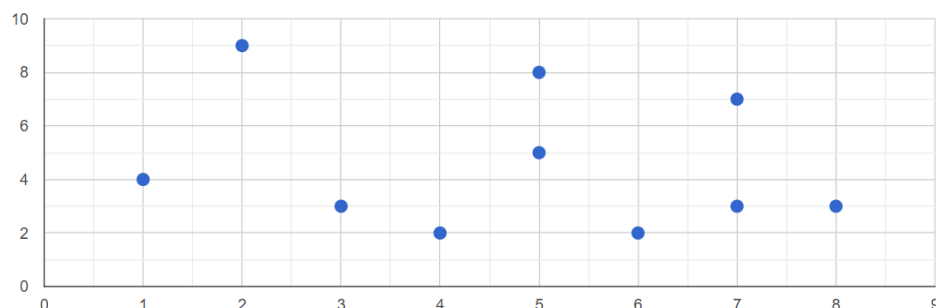
### Διευκρινίσεις:

- Επιτρέπεται η χρησιμοποίηση έτοιμου κώδικα ελεύθερα διαθέσιμου (έτοιμες υλοποιήσεις με κατάλληλη προσαρμογή από εσάς) με την προϋπόθεση ότι θα αναφέρεται σαφώς στα σχόλια η πηγή του. Όμως, τα προγράμματα θα ελέγχονται από αυτόματο σύστημα εντοπισμού αντιγραφών. Αν εντοπιστούν αντιγραφές μεταξύ φοιτητών, τότε οι φοιτητές αυτοί θα μηδενίζονται.
- Στο επάνω μέρος του αρχείου java με τον πηγαίο κώδικα να αναφέρονται οπωσδήποτε σε σχόλια το ονοματεπώνυμο, το ΑΕΜ και το ακαδημαϊκό email σας.
- Οι φοιτητές θα πρέπει να είναι έτοιμοι να δώσουν προφορικές εξηγήσεις για την υλοποίησή τους, εφόσον τους ζητηθεί.
- Οι βαθμοί των ασκήσεων θα ισχύουν και για τις εξετάσεις Σεπτεμβρίου ή επί πτυχίω.

### Μερικά ακόμα test cases:

Uniformly distributed:

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

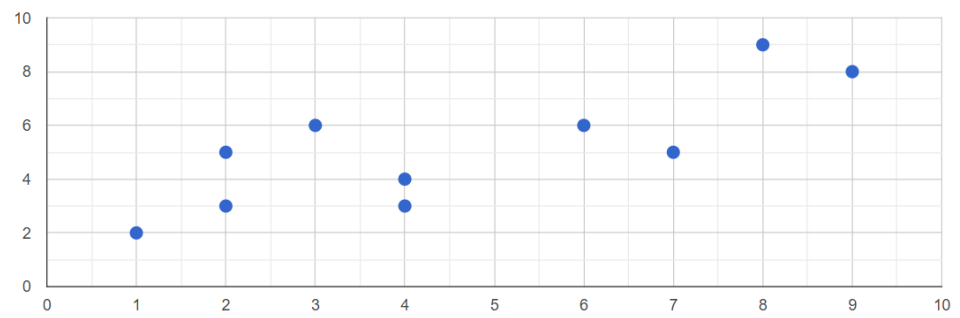


Skyline:

1 4
3 3
4 2

Correlated:

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

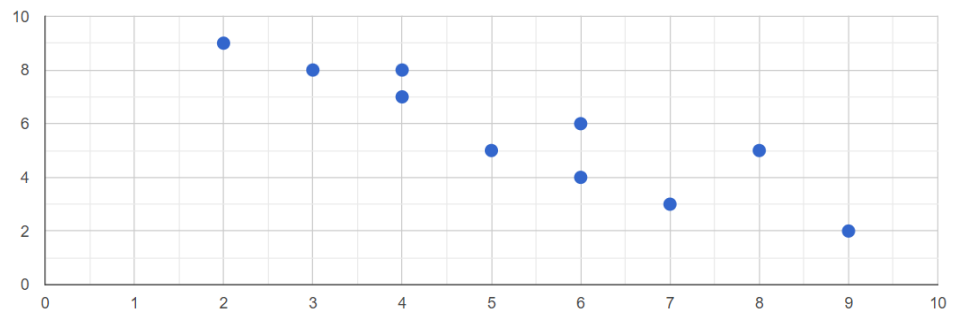


Skyline:

1 2
-----

Anti-Correlated:

10
9 2
8 5
6 4
7 3
5 5
6 6
4 7
4 8
3 8
2 9



Skyline:

2 9
3 8
4 7
5 5
6 4
7 3
9 2