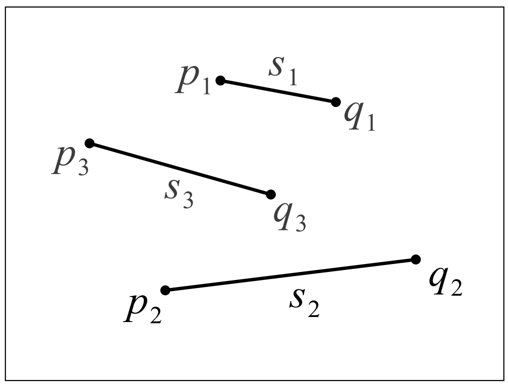
Υπολογιστική Γεωμετρία
Άσκηση 2η

Ερώτημα 1.
Θεωρήστε τον αυξητικό αλγόριθμο κατασκευής τραπεζοειδών χαρτών και εντοπισμού σημείου. Θεωρήστε τα τμήματα που εμφανίζονται στο παρακάτω σχήμα και υποθέστε ότι τα τμήματα εισάγονται με τη σειρά (s_1, s_2, s_3).

(α) Δείξτε τον τραπεζοειδή χάρτη και τη δομή αναζήτησης μετά την εισαγωγή των s_1 και s_2.

(β) Δείξτε τον τραπεζοειδή χάρτη και τη δομή αναζήτησης μετά την εισαγωγή και των τριών τμημάτων.



Ερώτημα 2.
Έστω P και Q δύο σύνολα σημείων στο επίπεδο μεγέθους n και m αντίστοιχα με $n \leq m$. Δώστε έναν αποδοτικό αλγόριθμο ο οποίος για κάθε σημείο $q \in Q$ να υπολογίζει το πλησιέστερο σημείο του q μεταξύ των σημείων του P. Αναλύστε τον χρόνο εκτέλεσης του αλγορίθμου σας.

Ερώτημα 3.
Έστω S ένα σύνολο n ευθύγραμμων τμημάτων στο επίπεδο. Σχεδιάστε έναν αλγόριθμο $O(n^2)$ χρόνου που να αποφασίζει αν υπάρχει μία ευθεία ℓ η οποία να τέμνει όλα τα τμήματα του S.

Ερώτημα 4.
Έστω P και Q δύο σύνολα σημείων στο επίπεδο με $|P| = |Q| = n$. Δώστε έναν αλγόριθμο ο οποίος να αποφασίζει σε αναμενόμενο χρόνο $O(n)$ εάν υπάρχει ευθεία ℓ η οποία να διαχωρίζει τα σημεία των P και Q μεταξύ τους.

Υπολογιστική Γεωμετρία

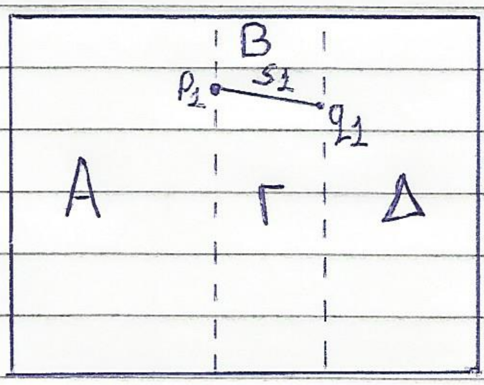
Άσκηση 2η

Αλέξανδρος Πλέσσιας
2025201100068
cst11068@uop.gr

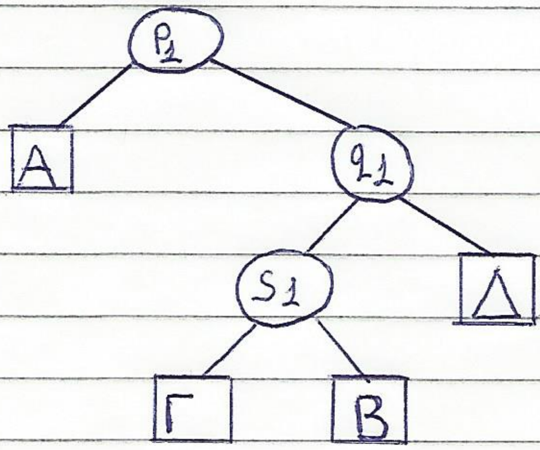
Ερώτημα 1 Για την επίλυση αυτής της ερώτησης βασίστηκα στο "enot7-entopismos-shmeiou.pdf" (Σελίδες 32-38) και στο "cg-notes-2015-16.pdf" (Σελίδες 50-51).

(α) Αρχικά εισάγω το ευθύγραμμο τμήμα s_1.

Τραπεζοειδής χάρτης:

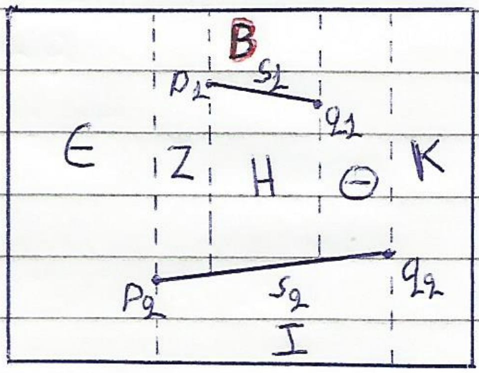


Δομή αναζήτησης:

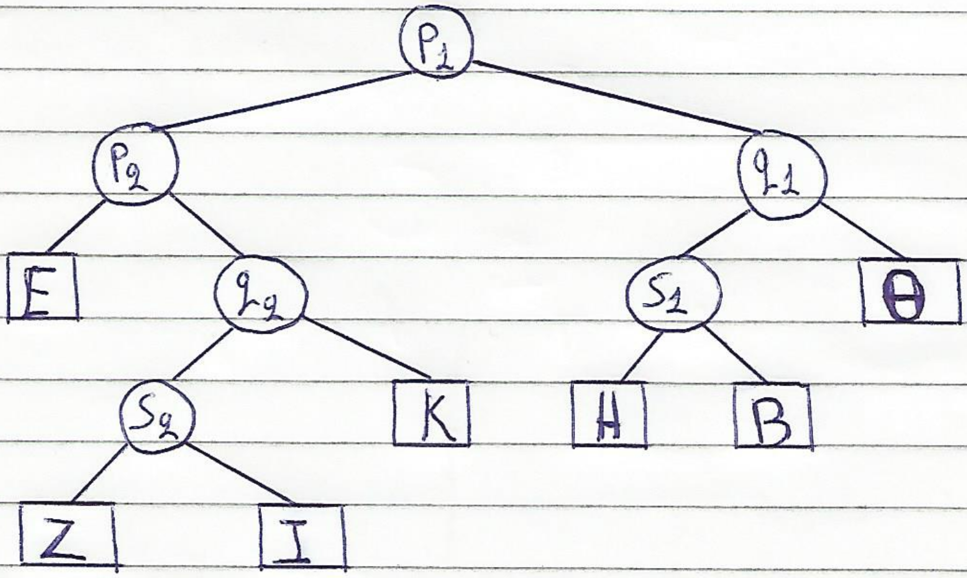


Στη συνέχεια εισάγω το ευθύγραμμο τμήμα s_2.

Τραπεζοειδής χάρτης:

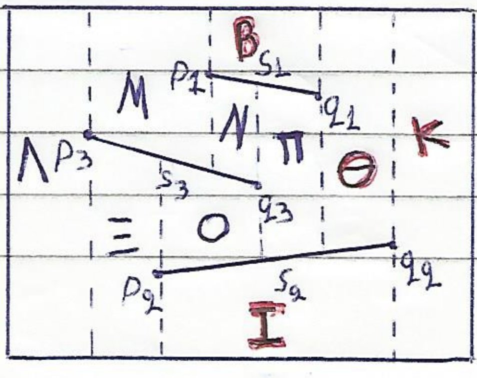


Δομή αναζήτησης:

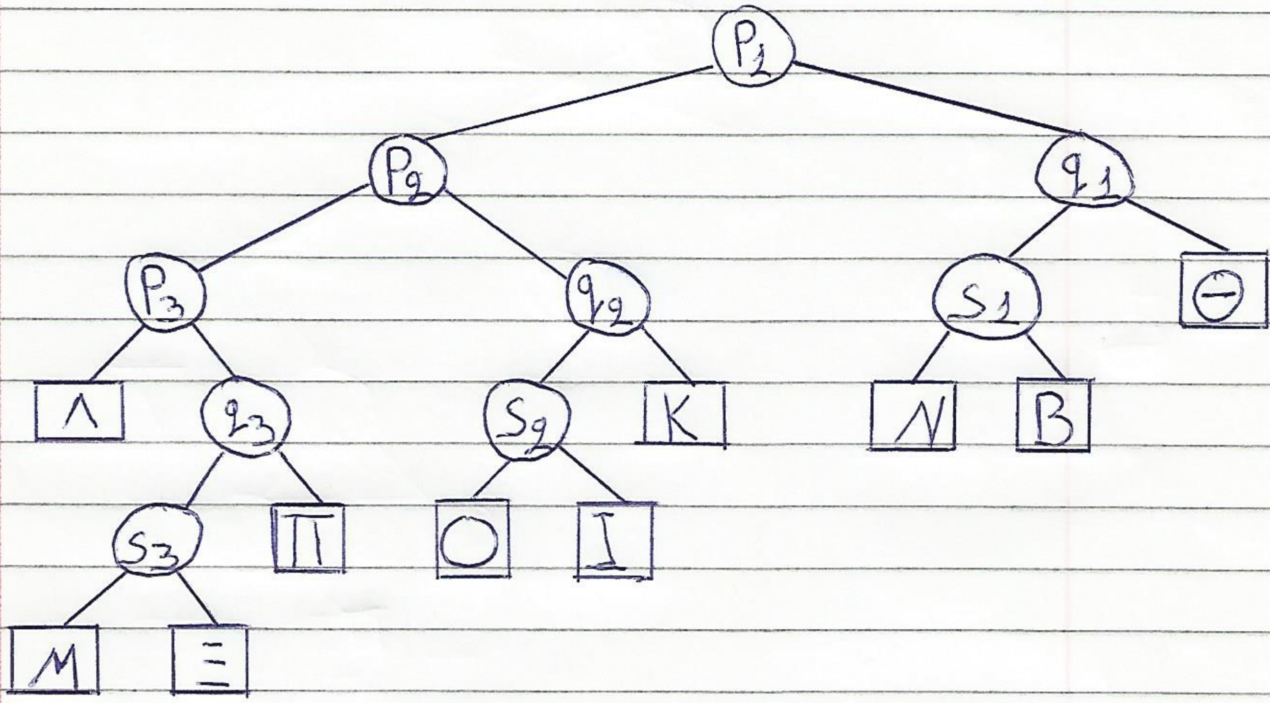


(B) Τέλος, εισάγω το 3ο και τελευταίο ευθύγραμμο τμήμα s_3.

Τραπεζοειδής χάρτης:



Δομή αναζήτησης:



Ερώτημα 2: Ουσιαστικά μας ζητάει να περιγράψουμε πως [redacted] λύνεται η εφαρμογή "Ερωτήσεις πλησιέστερου γείτονα" (cg-notes-2015-16.pdf [redacted] σελίδα 56). Οπότε αρχικά θα υπολογίσω ένα διάγραμμα Voronoi του συνόλου P [redacted] με την βοήθεια του αλγόριθμου Fortune και έπειτα θα τρέξω για κάθε σημείο του συνόλου Q τον αλγόριθμο εντοπισμού σημείου.

Χρόνος εκτέλεσης: Ο αλγόριθμος Fortune για το σύνολο σημείων P χρειάζεται χρόνο $O(n\log n)$ και ο αλγόριθμος εντοπισμού σημείου χρειάζεται χρόνο $O(\log n)$ χρόνο αλλά θα χρειαστεί να εκτελεστεί για κάθε σημείο του συνόλου Q άρα m φορές. Οπότε ο συνολικός χρόνος θα είναι $O(n\log n) + O(m\log n)$ και επειδή από την εκφώνηση δίνεται ότι $n \leq m$, ο τελικός χρόνος εκτέλεσης θα είναι $O(m\log n)$.

Ερώτημα 3 Αρχικά εφαρμόζω δυισμό, δηλαδή όλα τα σημεία όλων των ευθύγραμμων τμημάτων του συνόλου S μεταφέρονται στο δυϊκό επίπεδο, αυτή η διαδικασία έχει χρονική πολυπλοκότητα $2n * O(1)$ που ισοδυναμεί με $O(2n)$, οπού τελικά είναι $O(n)$. Έπειτα χρειάζομαι να βρω ένα σημείο, εάν υπάρχει, το οποίο να ανοική σε όλα τα ευθύγραμμα τμήματα του συνόλου S, δηλαδή τα ημί-επίπεδα των περιοχών όλων των σημείων του συνόλου S στο δυϊκό επίπεδο. Το σημείο που θα βρω, εάν υπάρχη θα αντιστοιχή στην ευθεία l που ζητάται στην εκφώνηση σας. Για να βρω το σημείο αυτό θα χρησιμοποιήσω τον αλγόριθμο σάρωσης επιπέδου στο δυϊκό επίπεδο. Θα κρατάω όμως μόνο τα γεγονότα που έχουν να κάνουν με σημεία τομής ευθειών για να έχω χρόνο εκτέλεσης $O(2n)$ που ισοδυναμεί με $O(n)$, η πληροφορία για τον χρόνο εκτέλεσης σε $O(n)$ χρόνο αναφέρεται στο "cg-notes-2015-16.pdf" στη σελίδα 19. Οπότε επειδή το επίπεδο μας είναι δυϊκό (μιας και κάθε ευθύγραμμο τμήμα έχει 2 σημεία άρα 2 ευθείες) ο χρόνος υπολογισμού των τομών μέσω του αλγόριθμου σάρωσης επιπέδου/επιπέδων θα γίνει $O(n^2)$ που είναι και ο ζητούμενος.

Πότε υπάρχει ευθεία l: Για να υπάρχει ευθεία l στο πρωτεύων επίπεδο θα πρέπει να υπάρχη το σημείο που αναφέραμε παραπάνω. Το σημείο αυτό ξέρουμε ότι υπάρχει όταν βρίσκεται από πάνω και από κάτω, από την υψηλότερη ευθεία που ορίζει το ημιεπίπεδο από πάνω της και τη χαμηλότερη ευθεία που ορίζει το >> >> κάτω της αντίστοιχα.

Παρατήρηση: Δεν χρησιμοποίησα καθόλου όπως βλέπετε γραμμικό προγραμματισμό, ακριβώς όπως μας προτείνατε στην τάξη.

Ερώτημα 4 Αρχικά εφαρμόζω δυϊσμό για όλα τα σημεία των συνόλων P και Q, αυτή η διαδικασία έχει χρονική πολυπλοκότητα $2n \cdot O(1)$ δηλαδή $O(2n)$ που είναι ισοδύναμο με $O(n)$. Έπειτα για να βρώ την ευθεία l θα χρησιμοποιήσω γραμμικό προγραμματισμό και ουσιαστικά οι ευθείες των σημείων P και Q στο δυϊκό επίπεδο αναπαριστούν περιορισμούς οπότε θα κάνω μια υπόθεση εάν τα σημεία Q είναι πάνω από τα σημεία P, τότε θα πολ/ω με -1 τις ευθείες/ανισότητες του Q (για να τις αντιστρέψω) για να εξετάσω την περιοχή που είναι κάτω από τις ευθείες (minimize - εφικτή περιοχή) χρησιμοποιώντας τον πιθανοτικό αλγόριθμο που έχει αναμενόμενο χρόνο εκτέλεσης $O(n)$, υπάρχει στο "cg-notes-2015-16.pdf" στην σελίδα 89, εάν ο αλγόριθμος επιστρέψει εφικτή λύση τότε τερματίζει. Εάν όμως επιστρέψει μη-εφικτή λύση τότε θα πρέπει η υπόθεση να αλλάξει δηλαδή ότι τα σημεία P είναι πάνω από τα σημεία Q, τότε θα πολ/ω με -1 τις ευθείες/ανισώσεις του P για να εξετάσω πάλι την περιοχή που είναι κάτω από τις ευθείες, πάλι με την βοήθεια του πιθανοτικού αλγόριθμου, εάν επιστρέψει εφικτή λύση τότε υπάρχει ευθεία l, εάν επιστρέψει πάλι μη-εφικτή λύση τότε δεν υπάρχει καμία ευθεία l. Οπότε όπως είναι προφανές ο χρόνος εκτέλεσης στην χειρότερη είναι $(2 * O(n) + O(n))$ που ισοδύναμη με το $O(n)$ που ζητάει στην εκφώνηση εκς.