Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Αριστείδης Παγουρτζής, Δώρα Σουλίου, Ευστάθιος Ζάχος, Δημήτρης Σακαβάλας
Λύσεις 1ης Σειράς Προγραμματιστικών Ασκήσεων

Άσκηση 1: Συνάντηση στην Τετραγωνούπολη

Μια πρώτη λύση που μπορούμε να κάνουμε είναι η διαδοχική δοκιμή κάθε σημείο και ο υπολογισμός των αποστάσεων από όλα τα υπόλοιπα. Για κάθε σημείο χρειαζόμαστε γραμμικό χρόνο για τον υπολογισμό και συνεπώς η πολυπλοκότητα που προκύπτει είναι $\Theta(N^2)$ η οποία δεν επαρκεί για όλα τα αρχεία ελέγχου.

Ας συμβολίσουμε ως $A = [(x_1, y_1), ..., (x_N, y_N)]$ τον πίνακα όπου κρατάμε τις συντεταγμένες των σημείων. Κατασκευάζουμε δύο νέους πίνακες $A_X = [(x_1, 1), ..., (x_i, i), ..., (x_N, N)]$ και $A_Y = [(y_1, 1), ..., (y_i, i), .., (y_N, N)]$ οι οποίοι, όπως φαίνεται, κρατάνε ξεχωριστά τις συντεταγμένες του A χωριστά, καθώς και το index του σημείου που έχει την αντίστοιχη συντεταγμένη στον πίνακα A .

Ταξινομούμε τον κάθε πίνακα και επεξεργαζόμαστε κάθε σημείο ξεχωριστά, όπως και στην πρώτη λύση, με την διαφορά πως επεξεργαζόμαστε τα σημεία με την σειρά που εμφανίζονται στον ταξινομημένο πίνακα A_X. Έστω πως θέλουμε να υπολογίσουμε την απάντηση για κάποιο σημείο με συντεταγμένες (x, y) το οποίο εμφανίζεται στην θέση i του πίνακα A_X . Υπολογίζουμε πρώτα την συνεισφορά των X-συντεταγμένων στην συνολική απόσταση, την οποία συμβολίζουμε d_{X_i}.

$$\begin{aligned} d_{X_i} &= \sum_{k=1}^{N} |A_X[i] - A_X[k]| = \\ &= \sum_{k=1}^{i} |A_X[i] - A_X[k]| + \sum_{k=i+1}^{N} |A_X[i] - A_X[k]| \end{aligned} \tag{1}$$

Επειδή ο πίνακας A_X είναι ταξινομημένος σε αύξουσα σειρά έχουμε

$$\begin{aligned} d_{X_i} &= \sum_{k=1}^{i-1} (A_X[i] - A_X[k]) + \sum_{k=i+1}^{n} (A_X[k] - A_X[i]) = \\ &= (i-1)A_X[i] - \sum_{k=0}^{i-1} A_X[k] - (N-i)A_X[i] + \sum_{k=i+1}^{N} A_X[k] \end{aligned} \tag{2}$$

Υπολογίζουμε τον πίνακα μερικών αθροισμάτων $S[i] = \sum_{k=1}^{i} A[i]$ σε O(N) και η παραπάνω σχέση γίνεται:

$$d_{X_i} = (2i - N - 1)A_X[i] - S[i-1] + S[N] - S[i] \tag{3}$$

Το οποίο υπολογίζεται σε O(1) για κάθε σημείο.

Αφού υπολογίσουμε την συνεισφορά της X-συντεταγμένης, βρίσκουμε την Ψ-συντεταγμένη του σημείου i, κοιτώντας τον πίνακα A και με μία δυαδική αναζήτηση βρίσκουμε την Υ-συντεταγμένη στον πίνακα A_Y. Έπειτα ακολουθούμε ακριβώς την ίδια λογική που ακολουθήσαμε παραπάνω για τον υπολογισμό του d_y και αθροίζουμε τις δύο συνεισφορές.

Έτσι, για κάθε σημείο καταφέρνουμε σε $\Theta(logN)$ χρόνο, κάνοντας αρχικά δύο ταξινομήσεις, οπότε η συνολική πολυπλοκότητα του αλγορίθμου μας είναι $\Theta(NlogN)$.

Bonus: Έπειδή τα όρια των συντεταγμένων είναι σχετικά μικρά, $X, Y \leq M \leq 10^7$, μπορούμε να κάνουμε Counting-Sort αντί για παραδοσιακή συγκριτική ταξινόμηση και να αντικαταστήσουμε την δυαδική αναζήτηση με ένα look-up σε πίνακα συχνοτήτων μεγέθους M. Έτσι, θα έχουμε πολυπλοκότητα $\Theta(N + M)$.

Άσκηση 2: Πόλεμος στην χώρα των Αλγορίθμων

Είναι εύκολο να πετύχουμε πολυπλοκότητα της τάξης του $\Theta(N^2)$, απλά υπολογίζοντας όλες τις συγκρούσεις των σωματιδίων και με προσοχή επιλέξουμε τις πρώτες K. Για την λύση του προβλήματος, χωρίζουμε την διαδικασία σε K γύρους, σκοπός μας είναι κάθε φορά να επιλέγουμε την πρώτη σύγκρουση και να αφαιρούμε τα δύο σωματίδια που μετέχουν σε αυτή. Συνεπώς, χωρίς βλάβη της γενικότητας, θεωρούμε από εδώ και στο εξής ότι μας ενδιαφέρει να βρούμε την πρώτη σύκρουση.

Παρατηρούμε το εξής, αν κάποια χρονική στιγμή παρατηρήσουμε τις θέσεις των σωματιδίων και δούμε πως υπάρχει κάποιο σωματίδιο-α δεξιά κάποιου σωματίδιου-β τότε ξέρουμε πως κάποια στιγμή στο παρελθόν έγινε κάποια σύγκρουση. Αντίστροφα, αν όλα τα σωματίδια-α βρίσκονται αριστερά όλων των σωματιδίων-β ξέρουμε πως δεν έχει συμβεί σύγκρουση μέχρι στιγμής.

Έστω πως θέλαμε να ελέγχουμε για κάποιον δεδομένο χρόνο T, αν έχει γίνει κάποια σύγκρουση στο διάστημα (0, T] τότε, όπως είπαμε και προηγουμένως, αρκεί να υπολογίσουμε τις θέσεις όλων των σωματιδίων την χρονική στιγμή T σε γραμμικό χρόνο και να ελέγξουμε αν ισχύει η πρώτη συνθήκη που αναφέραμε. Εφ΄ όσον το έχουμε αυτό, μένει να κάνουμε μια δυαδική αναζήτηση στον χρόνο T και να χρησιμοποιούμε την διαδικασία που περιγράψαμε για να αποφανθούμε αν θα κινηθούμε κάθε φορά στο παρελθόν ή στο μέλλον. Συγκεκριμένα, αν για κάποια χρονική στιγμή T_1 δεν έχει γίνει σύγκρουση, ξέρουμε πως θα γίνει κάποια στιγμή μετά την T_1 και αντίστροφα.

Η πολυπλοκότητα κάθε γύρου είναι $\Theta(NlogL)$, αφού ο μέγιστος χρόνος κάποιας σύγκρουσης δεν μπορεί να ξεπερνάει το L. Συνεπώς η συνολική μας πολυπλοκότητα είναι $\Theta(KNlogL)$.

Bonus: Για να ρίξουμε την πολυπλοκότητα σε $\Theta(NK)$ αναπαριστούμε τις κινήσεις των σωματιδίων σαν ευθείες στο επίπεδο. Συγκεκριμένα για τα σωματίδια α έχουμε $x = v_i t - v_i t_i$, Ενώ για τα σωματίδια τύπου β έχουμε: $x = -v_i t + L + v_i t_i$.

Από αυτές τις ευθείες κρατάμε το convex hull , όπως περιγράφεται στο (1). Κρατάμε τις ευθείες που μετέχουν στο convex hull σε αύξουσα σειρά κλίσης. Πλέον, το ζητούμενο του προβλήματος μας έχει μετασχηματιστεί στο εξής: Δεδομένων 2 συνόλων ευθειών, θέλουμε να βρούμε τις 2 ευθείες, που να μην ανήκουν στο ίδιο σύνολο, τέτοιες ώστε το σημείο τομής τους να έχει την μικρότερη δυνατή τετμημένη.

Για να το βρούμε αυτό σε γραμμικό χρόνο, εφαρμόζουμε την τεχνική των δύο δεικτών. Ξεκινώντας 2 δείκτες που δείχνουν στην πρώτη ευθεία και των δύο συνόλων, ξεκινάμε να αυξάνουμε τον πρώτο, όσο το σημείο τομής των ευθείων μετατοπίζεται αριστερά. Αν κάποια στιγμή δούμε πως το σημείο τομής για την επόμενη θέση του πρώτου δείκτη έχει μεγαλύτερη τετμημένη από την τωρινή θέση, προχωράμε τον άλλον δείκτη. Ο αλγόριθμος αυτός είναι γραμμικού χρόνου και επαναλαμβάνεται K φορές, συνεπώς η πολυπλοκότητα του αλγορίθμου είναι $\Theta(NK)$.

References:

1. https://wcipeg.com/wiki/Convex_hull_trick