**ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ**

**ΕΡΓΑΣΙΑ 2Η (έως δύο άτομα)**

**ΒΑΡΥΤΗΤΑ 50% ΣΤΗΝ ΤΕΛΙΚΗ ΒΑΘΜΟΛΟΓΙΑ**

Στην εργασία αυτή καλείσθε να επιλύσετε ένα σύνολο ασκήσεων που αναφέρονται σε όλη τη διδαχθείσα ύλη. Όπου απαιτείται αλγόριθμος θα πρέπει στην απάντησή σας να περιλαμβάνεται ψευδοκώδικας ο οποίος θα συνοδεύεται από αναλυτική τεκμηρίωση. Ιδιαίτερη έμφαση θα πρέπει να δοθεί στην απόδειξη της ορθότητας των αλγορίθμων και της σχετικής χρονικής πολυπλοκότητας. Όλες οι ασκήσεις έχουν την ίδια βαρύτητα στην συνολική βαθμολογία της εργασίας. Η παράδοση της εργασίας θα γίνει με την αποστολή ενός αρχείου pdf στην πλατφόρμα gunet. Η προθεσμία παράδοσης της εργασίας ορίζεται η Πέμπτη 29/2/2024.

**1.** Γράψτε ένα πρόγραμμα που θα διαβάζει $N$ σημεία στο επίπεδο και θα επιστρέφει ως έξοδο οποιοδήποτε ομάδα από τέσσερα ή περισσότερα συγγραμμικά σημεία (δηλ. σημεία στην ίδια ευθεία). Ο προφανής αλγόριθμος απαιτεί χρόνο $O(N^4)$. Σχεδιάστε ένα αλγόριθμο που επιλύει το συγκεκριμένο πρόβλημα και βασίζεται στην ταξινόμηση και έχει χρόνο εκτέλεσης $O(N^2\log N)$.

**2.** Όταν υλοποιούμε τον αλγόριθμο Quicksort, αν ένας πίνακας περιέχει πολλά ίσα στοιχεία, μπορεί να είναι καλύτερο να εκτελέσουμε μία διαμέριση σε τρία μέρη (σε στοιχεία μικρότερα από, ίσα με και μεγαλύτερα από το στοιχείο οδηγό), προκειμένου να εκτελέσουμε λιγότερες αναδρομικές κλήσεις. Θεωρείστε ότι το σύστημα προσφέρει μία λειτουργία compare που ελέγχει αν δύο στοιχεία είναι ίσα ή ποιο στοιχείο είναι μεγαλύτερο από το άλλο.

**α.** Δώστε ένα αλγόριθμο που εκτελεί $N$-1 λειτουργίες compare για τη δημιουργία της παραπάνω διαμέρισης. Επίσης, προσπαθήστε να ελαχιστοποιήσετε το πλήθος των μετακινήσεων στοιχείων των στοιχείων που θα χρειαστείτε για την επίτευξη της διαμέρισης.

**β.** Αποδείξτε ότι χρησιμοποιώντας τον παραπάνω αλγόριθμο, η ταξινόμηση ενός πίνακα $N$ στοιχείων που περιέχει μόνο d διαφορετικές τιμές απαιτεί χρόνο O(dN).

**3.** Ας υποθέσουμε ότι πρέπει να βρούμε τα $k$ μικρότερα στοιχεία σε μία λίστα $N$ στοιχείων και δεν μας ενδιαφέρει η σχετική τους σειρά. Μπορεί να βρεθεί αλγόριθμος γραμμικού χρόνου όταν το $k$ είναι σταθερά; Αιτιολογήστε την απάντησή σας. Δώστε ένα αλγόριθμο χρόνου $O(N+k\log k)$ όταν τα $k$ στοιχεία πρέπει να ληφθούν ταξινομημένα.

**4.** Έστω ότι έχουμε δύο πίνακες A και B $N$ ακεραίων, πιθανώς με διπλότυπα, στο διάστημα από 1 ως 2$N$. Περιγράψτε ένα αλγόριθμο γραμμικού χρόνου που διαπιστώνει αν οι A και B περιέχουν το ίδιο σύνολο στοιχείων (πιθανώς με διαφορετική σειρά).

**5.** Δοθέντος ενός συνόλου S $N$ σημείων στο επίπεδο, απαιτείται η εύρεση του συνόλου $M\subseteq S$ των μέγιστων σημείων του. Ένα σημείο $p=(p_x,p_y)$ καλείται μέγιστο (maximal), εάν δεν «κυριαρχείται» (dominated) από κανένα άλλο στοιχείο, δηλ., δεν υπάρχει άλλο σημείο $q=(q_x,q_y)$ στο S, τέτοιο ώστε $p_x\leq q_x$ και $p_y\leq q_y$. Δώστε ένα αλγόριθμο βασισμένο στην τεχνική Διαίρει και Βασίλευε που επιλύει αυτό το πρόβλημα.

**6.** Έστω πίνακας A $N$ διαφορετικών στοιχείων με το χαρακτηριστικό ότι υπάρχει μία θέση, i, τέτοια ώστε η ακολουθία A[0],A[1],…,A[$i$] είναι αύξουσα, ενώ η ακολουθία A[$i$],A[$i$+1],…,A[$N$-1] είναι φθίνουσα. Να γραφεί και να αναλυθεί αλγόριθμος με βάση την τεχνική Διαίρει και Βασίλευε που εντοπίζει τη θέση $i$.

**7.** Έστω ότι ένας περιοδεύων αντιπρόσωπος πωλήσεων έχει προγραμματίσει $N$ επισκέψεις στις ημερομηνίες $d_1,d_2,\dots,d_N$ και το κόστος των ταξιδιών σε βενζίνη είναι αντίστοιχα $c_1,c_2,\dots,c_N$. Εάν αγοράσει ένα εκπτωτικό κουπόνι βενζίνης, με κόστος $r$, τότε για τον επόμενο χρόνο θα έχει έκπτωση $e$% κάθε φορά που βάζει βενζίνη. Με τη βοήθεια δυναμικού προγραμματισμού, να βρεθεί πότε πρέπει να αγορασθεί το εκπτωτικό κουπόνι προκειμένου να ελαχιστοποιηθεί το συνολικό κόστος των ταξιδιών.

**8.** Έστω ένα κομμάτι μαρμάρου διαστάσεων $M$x$N$. Υποθέστε ότι σας δίνεται ένας πίνακας K διαστάσεων $M$x$N$, όπου η θέση K[$i$][$j$] δίνει το κέρδος του μαρμαρά ένα πουλήσει ένα μάρμαρο διαστάσεων $i$x$j$, όπου $i$ και $j$ είναι θετικοί ακέραιοι. Δεδομένου ότι ο τεμαχισμός γίνεται μόνο οριζόντια και κάθετα, να γραφτεί αλγόριθμος με βάση τον δυναμικό προγραμματισμό που υπολογίζει το μέγιστο κέρδος που μπορεί να δώσει το κομμάτι $M$x$N$ με κατάλληλο τεμαχισμό.

**9.** Έστω S={$t_1,t_2,\dots,t_N$} ένα σύνολο $N$ εργασιών, όπου κάθε εργασία $t_i=(t_i^1,t_i^2)$ απαρτίζεται από δύο υποέργα με διαφορετική διάρκεια $d_i^1,d_i^2$, ενώ η πρώτη υποεργασία $t_i^1$ πρέπει να εκτελεσθεί πριν από την έναρξη της δεύτερης $t_i^2$. Οι εργασίες $t_i$ δεν εμπλέκονται μεταξύ τους. Με δεδομένο ότι για τα υποέργα $d_i^1$ υπάρχει μόνο ένας αποκλειστικός πόρος $r$, ενώ για τα υποέργα $d_i^2$ υπάρχουν διαθέσιμοι $N$ άλλοι πόροι, να βρεθεί μία δρομολόγηση που ελαχιστοποιεί τον χρόνο ολοκλήρωσης του S.

**10.** Έστω ένα γράφημα $G(V,E)$ με βάρη στις ακμές και έστω $T$ ένα ελάχιστο γεννητικό δέντρο του $G$.

**α.** Σχεδιάστε και αναλύστε ένα αλγόριθμο ενημέρωσης του $T$ στην περίπτωση που προσθέσουμε ή αφαιρέσουμε μία ακμή.

**β.** Για κάθε ακμή $e$ στο $E$ να βρεθεί το εύρος τιμών που μπορεί να πάρει το βάρος του, ώστε το $T$ να εξακολουθεί να είναι ελάχιστο γεννητικό δέντρο.

**11.** Προτείνετε ένα αλγόριθμο που βρίσκει το ελάχιστο από όλα τα γεννητικά δέντρα ενός γραφήματος (με βάρη στις ακμές) τα οποία περιέχουν μία συγκεκριμένη ακμή $e=(v,w)$.

**12.** Έστω ένα γράφημα $G(V,E)$ με θετικά βάρη στις ακμές. Περιγράψτε και αναλύστε ένα αλγόριθμο υπολογισμού του συντομότερου μονοπατιού από μία κορυφή $v$ σε μία κορυφή $w$, αποτελούμενο από $k$ ακριβώς ακμές.

**13.** Υποθέστε ότι ακμές ενός γραφήματος έχουν θετικά βάρη ενώ επιπλέον κάθε ακμή είναι χρωματισμένη είτε άσπρη είτε μαύρη. Σχεδιάστε και αναλύστε ένα αλγόριθμο, οποίος βρίσκει το συντομότερο μονοπάτι από μία κορυφή $v$ σε μία άλλη κορυφή $w$, με τον επιπλέον περιορισμό ότι πρώτα πρέπει να εμφανίζονται οι άσπρες ακμές και μετά να ακολουθούν οι μαύρες.