

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΑΣΚΗΣΕΙΣ ΑΛΓΟΡΙΘΜΟΙ: ΚΕΦΑΛΑΙΟ 2^ο

Ακαδημαϊκό έτος: 2014-2015

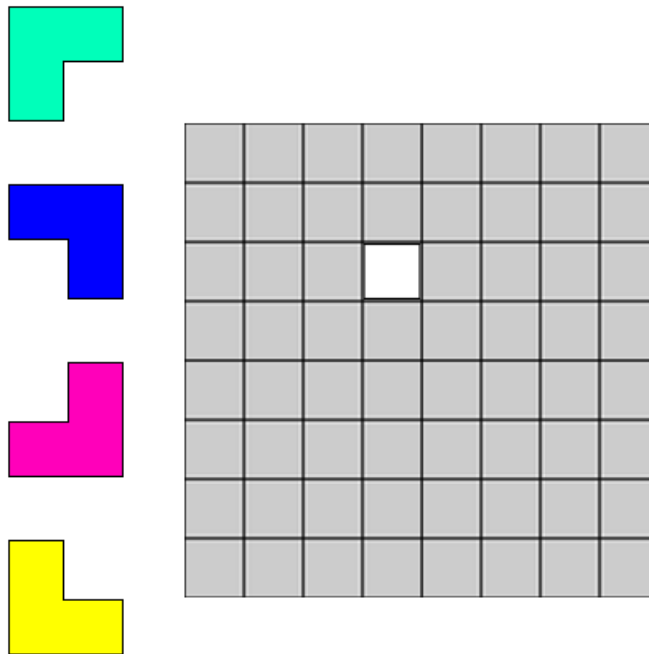
Διδάσκων: Χ. Κωνσταντόπουλος

1. Επιλύστε την ακόλουθη αναδρομική σχέση:

$$f(n) = \begin{cases} d & \text{if } n = 1 \\ 2f(n/2) + bn \log n & \text{if } n \geq 2 \end{cases}.$$

όπου d και b μη αρνητικές σταθερές και ο αριθμός n είναι δύναμη του 2.

2. Tromino καλείται ένα κομμάτι σε σχήμα L αποτελούμενο από 3 τετράγωνα μεγέθους 1×1 . Ζητούμενο είναι η κάλυψη ενός τετράγωνου πλέγματος μεγέθους $2^n \times 2^n$, από το οποίο λείπει ένα τετράγωνο κελί, με κομμάτια tromino. Κάθε tromino μπορεί να τοποθετηθεί με οποιονδήποτε τρόπο, αλλά θα πρέπει συνολικά να καλύπτονται όλα τα κελιά του τετράγωνου πλέγματος, εκτός από αυτό που απουσιάζει, χωρίς τυχούσες επικαλύψεις.



3. Έστω n σημεία στον ευκλείδειο χώρο. Χρησιμοποιώντας την τεχνική του διαίρει και βασίλευε, αναπτύξτε έναν αλγόριθμο που θα εντοπίζει το ζευγάρι σημείων με τη μικρότερη απόσταση.
4. Έστω πίνακας A , μεγέθους n . Ένα στοιχείο του πίνακα A καλείται πλειοψηφόν στοιχείο (majority element) εάν εμφανίζεται περισσότερες από $n/2$ φορές.
Ακολουθώντας την τεχνική του διαίρει και βασίλευε, σχεδιάστε και αναλύστε έναν αλγόριθμο, ο οποίος θα εντοπίζει το στοιχείο πλειοψηφίας (majority element) σε χρόνο $O(n \log n)$.
Σημείωση: Έχετε στη διάθεσή σας μόνο την πράξη της ισότητας και δεν μπορείτε να χρησιμοποιήσετε κάποιο αλγόριθμο ταξινόμησης.
5. Έστω πίνακας A , μεγέθους n . Ακολουθώντας την τεχνική του διαίρει και βασίλευε, σχεδιάστε και αναλύστε έναν αλγόριθμο, ο οποίος θα εντοπίζει στοιχεία τα οποία εμφανίζονται περισσότερες από $n/3$ φορές, σε χρόνο $O(n \log n)$.

6. Προτείνετε ένα αλγόριθμο για την εύρεση της κυρτής θήκης n σημείων στο επίπεδο, χρησιμοποιώντας την τεχνική του διαίρει και βασίλευε.
7. Έστω πίνακας A , μεγέθους n . Σχεδιάστε και αναλύστε αλγόριθμο ο οποίος εντοπίζει το k -οστο μεγαλύτερο στοιχείο του πίνακα, με πολυπλοκότητα μέσου όρου $O(n)$ και ένα δεύτερο αλγόριθμο με πολυπλοκότητα χειρότερης περίπτωσης $O(n)$.

8. Έστω πίνακας A , μεγέθους n . Ο A ονομάζεται k -ταξινομημένος εάν υποδιαιρείται σε k block, μεγέθους n/k , τέτοια ώστε τα στοιχεία κάθε block να είναι μεγαλύτερα από τα στοιχεία των προηγούμενων blocks και μικρότερα από τα στοιχεία των επόμενων blocks. Τα στοιχεία του κάθε block δεν χρειάζεται να είναι ταξινομημένα. Για παράδειγμα ο ακόλουθος πίνακας είναι 4-ταξινομημένος:

1	2	4	3	7	6	8	5	10	11	9	12	15	13	16	14
---	---	---	---	---	---	---	---	----	----	---	----	----	----	----	----

- α. Περιγράψτε έναν αλγόριθμο που υλοποιεί την k -ταξινόμηση ενός τυχαίου πίνακα σε χρόνο $O(n \log(n/k))$.
- β. Αποδείξτε ότι ένας αλγόριθμος k -ταξινόμησης που βασίζεται στη σύγκριση, χρειάζεται $\Omega(n \log(n/k))$ συγκρίσεις στη χειρότερη περίπτωση.
- γ. Περιγράψτε έναν αλγόριθμο που ταξινομεί πλήρως έναν k -ταξινομημένο πίνακα σε χρόνο $O(n \log k)$.
- δ. Αποδείξτε ότι ένας αλγόριθμος ταξινόμησης που βασίζεται στη σύγκριση για να ταξινομήσει πλήρως έναν k -ταξινομημένο πίνακα χρειάζεται $\Omega(n \log k)$ συγκρίσεις στη χειρότερη περίπτωση.
- Σημείωση: Σε κάθε περίπτωση υποθέστε ότι ο αριθμός n/k είναι ακέραιος.

9. Έστω k ταξινομημένες λίστες ίδιου μεγέθους και συνολικού μεγέθους n . Η συγχώνευση των λιστών δίνει μία ολική διάταξη. Αν συγχωνεύσουμε διαδοχικά τις λίστες, τι πολυπλοκότητα χρόνου θα έχει η πράξη αυτή; Περιγράψτε έναν αποδοτικότερο αλγόριθμο συγχώνευσης.

10. Έστω δύο ταξινομημένοι πίνακες A, B μεγέθους n έκαστος. Περιγράψτε και αναλύστε έναν αλγόριθμο, ο οποίος εντοπίζει το μεσαίο, από άποψη διατάξεως, στοιχείο των δύο πινάκων.

11. Έστω πίνακας ακεραίων A , μεγέθους n , με $A(i) \in \{0, \dots, n\}$, $n = 2^x - 1$, $x \in \mathbb{N}$, $i \in \{1, \dots, n\}$. Να αναπτυχθεί και να αναλυθεί αλγόριθμος ο οποίος εντοπίζει τον ακέραιο που λείπει, εντός γραμμικού χρόνου.

12. Έστω πίνακας A , μεγέθους n , ο οποίος περιέχει ανεξάρτητες ομοιόμορφα κατανομημένες τυχαίες τιμές. Περιγράψτε και αναλύστε αλγόριθμο ο οποίος ταξινομεί τον πίνακα A σε χρόνο $O(n)$ στη μέση περίπτωση και σε χρόνο $O(n \log n)$ στη χειρότερη περίπτωση.

13. Έστω το πρόβλημα της εύρεσης των ατόμων με κοινά ενδιαφέροντα-προτιμήσεις. Βασικό κομμάτι του προβλήματος είναι η σύγκριση δύο κατατάξεων (rankings). Έστω για παράδειγμα ότι κατατάσσουμε με βάση την προτίμηση μας ένα σύνολο n ταινιών και το ζητούμενο είναι να βρούμε άλλους χρήστες του διαδικτύου με παρόμοιες προτιμήσεις με τις δικές μας. Υπάρχει κάποια μετρική που να αξιολογεί την ομοιότητα δύο κατατάξεων από δύο διαφορετικούς χρήστες;

Ένας τρόπος να εκτιμήσουμε την ομοιότητα είναι να αριθμήσουμε τις ταινίες με αύξουσα σειρά με βάση την προτίμηση μας και να ελέγξουμε την ακολουθία των αριθμών που δημιουργείται με βάση την προτίμηση του άλλου χρήστη. Ο αριθμός των αναστροφών (inversions) ποσοτικοποιεί το βαθμό ομοιότητας των δύο κατατάξεων. Αν ισχύει $i < j$ και $A[i] > A[j]$, τότε το ζεύγος των (i, j) ονομάζεται ανάστροφο.

Προτείνετε και αναλύστε έναν αλγόριθμο που θα εκτιμά το βαθμό ομοιότητας δύο κατατάξεων μεγέθους n σε χρόνο $O(n \log n)$.