ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΑΣΚΗΣΕΙΣ
ΑΛΓΟΡΙΘΜΟΙ: ΚΕΦΑΛΑΙΟ 1$^{\text{ο}}$

Ακαδημαϊκό έτος: 2014-2015
Διδάσκων: Χ. Κωνσταντόπουλος

---

1. Συμπληρώστε την τάξη μεγέθους και δικαιολογείστε τις απαντήσεις σας.

$$nloglogn = [\quad](nlog^2n)$$
$$\sqrt[3]{nlogn} = [\quad](n^{0.3})$$
$$nlog(\sqrt[6]{n}) = [\quad](nlogn)$$
$$2^{\sqrt{2}}n = [\quad](2n^{1-\epsilon}, 0 < \epsilon < 1)$$
$$(logn)^{logn} = [\quad](n^{loglogn})$$
$$2^{2^n} = [\quad](n!)$$

2. Έστω ένα σύνολο $S = |n|$ σημείων του επιπέδου $\mathbb{R}^2$. Περιγράψτε έναν αποδοτικό αλγόριθμο για την εύρεση των μέγιστων σημείων του συνόλου S.
Σημείωση: Ένα σημείο $p = (p_x, p_y)$ καλείται μέγιστο (maximal), εάν δεν κυριαρχείται (dominated) από κανένα άλλο σημείο. Ισοδύναμα, δεν υπάρχει άλλο σημείο $q = (q_x, q_y) \in S$, τέτοιο ώστε $p_x \leq q_x$ και $p_y \leq q_y$.

3. Έστω πίνακας A, μεγέθους n, ο οποίος περιέχει πραγματικούς αριθμούς. Αναπτύξτε αλγόριθμο ο οποίος θα εντοπίζει τα στοιχεία του πίνακα A που είναι κυρίαρχα k θέσεις δεξιά και k θέσεις αριστερά τους. Για παράδειγμα το i-στό στοιχείο του πίνακα είναι κυρίαρχο αν είναι μεγαλύτερο ή ίσο με τα στοιχεία των θέσεων $j \in \{i-k, i-k+1, ..., i+k-1, i+k\}$.

4. Έστω το πρόβλημα αναζήτησης στοιχείου σε λίστα και γνωρίζουμε εκ των προτέρων την πιθανότητα αναζήτησης p_i ενός στοιχείου i που βρίσκεται στη θέση l_i της λίστας. Θα μπορούσαμε με κάποια αναδιάταξη της λίστας να μειώσουμε την πολυπλοκότητα της μέσης περίπτωσης του αλγορίθμου της γραμμικής αναζήτησης;
Στην περίπτωση που δεν γνωρίζουμε την πιθανότητα αναζήτησης των στοιχείων, υπάρχει κάποια τεχνική που να βελτιώνει την επίδοση του του αλγορίθμου της γραμμικής αναζήτησης στη μέση περίπτωση;
Αναλύστε και τεκμηριώστε τις απαντήσεις σας.

5. Κατά την είσοδο n πελατών σε ένα εστιατόριο, οι πελάτες αφήνουν στον υπάλληλο της υποδοχής τα παλτό τους. Κατά την αναχώρησή τους ο υπάλληλος τους τα επιστρέφει με τυχαία σειρά. Πόσοι πελάτες αναμένεται να λάβουν το δικό τους παλτό;

6. Έστω πίνακας A, μεγέθους n, με $A[i] \in \{1, 2, ..., n\}$ και $i \in \{1, 2, ..., n\}$. Αν ισχύει $i < j$ και $A[i] > A[j]$ τότε το ζεύγος των (i, j) ονομάζεται ανάστροφο. Έστω ότι τα στοιχεία του πίνακα είναι μία ομοιόμορφη μετάθεση των στοιχείων του συνόλου $\{1, 2, ..., n\}$. Πόσα ανάστροφα ζεύγη ενδέχεται να υπάρχουν στον πίνακα A;

7. Ποιο είναι το κάτω φράγμα του αριθμού των συγκρίσεων που πραγματοποιούνται στο πρόβλημα της έυρεσης του μεγίστου στοιχείου μιας μη διατεταγμένης λίστας;

8. Έστω το πρόβλημας της εύρεσης του μεγίστου και του ελαχίστου στοιχείου μιας μη διατεταγμένης λίστας. Βρείτε το άνω και το κάτω φράγμα του αριθμού των συγκρίσεων που χρειάζονται για την επίλυση του προβλήματος.

9. Έστω πίνακας A, μεγέθους n. Ο A ονομάζεται k-ταξινομημένος εάν για κάθε $i = 1, 2, ..., n-k$ ισχύει η ακόλουθη ανισότητα:

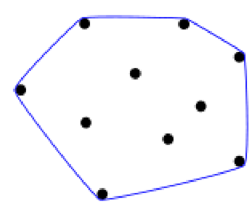
$$\frac{\sum_{j=i}^{i+k-1} A[j]}{k} \leq \frac{\sum_{j=i+1}^{i+k} A[j]}{k}$$

α. Τι ισχύει όταν ένας πίνακας είναι 1-ταξινομημένος;
β. Δώστε μία μετάθεση των αριθμών 1,2,...,10 η οποία είναι 2-ταξινομημένη, αλλά δεν είναι ταξινομένη.
γ. Αποδείξτε ότι ένας πίνακας A, μεγέθους n είναι k-ταξινομημένος αν και μόνο αν ισχύει:

$$A[i] \leq A[i+k] \text{ για κάθε } i = 1, 2, ..., n-k.$$

δ. Περιγράψτε έναν αλγόριθμο, ο οποίος ταξινομεί ένα πίνακα A, n στοιχείων, σε χρόνο $O(nlog(n/k))$.
ε. Δείξτε ότι ένας k-ταξινομημένος πίνακας μεγέθους n, μπορεί να ταξινομηθεί πλήρως σε χρόνο $O(nlogk))$.
στ. Δείξτε ότι όταν η παράμετρος k είναι σταθερά η k-ταξινόμηση ενός πίνακα n στοιχείων έχει πολυπλοκότητα $\Omega(nlogn)$.

10. Η κυρτή θήκη (convex hull) ενός συνόλου S, n σημείων στο επίπεδο $\mathbb{R}^2$, ορίζεται ως το μικρότερο κυρτό υποσύνολο του $\mathbb{R}^2$ που περιέχει το S.
 Υπολογίστε το κάτω φράγμα της χειρότερης περίπτωσης του προβλήματος της εύρεσης της κυρτής



θήκης ενός συνόλου S, n σημείων στο επίπεδο $\mathbb{R}^2$.

11. Δείξτε ότι δεν υπάρχει αλγόριθμος συγκριτικής διάταξης με γραμμική πολυπλοκότητα για τουλάχιστον τις μισές από τις $n/2$ πιθανές εισόδους μήκους n.

12. Έστω μία ταξινομημένη λίστα $X = x_1 < x_2 < ... < x_n$. Ζητούμενο είναι η αναζήτηση μιας τιμής y, όπου θα επιστρέφεται η τιμή 0 εάν η τιμή y δεν υπάρχει στη λίστα και η τιμή i εάν $y = x_i$. Για την επίλυση του προβλήματος έχουμε στη διάθεσή μας μόνο μία διαδικασία για την αναζήτηση στη λίστα, τη διαδικασία $TEST(i, j, k, y)$, η οποία επιστρέφει τις ακόλουθες απαντήσεις όταν ισχύει $0 < i < j < k \leq n$:
α. $y < x_i$
β. $y = x_i$
γ. $x_i < y < x_j$
δ. $y = x_j$
ε. $x_j < y < x_k$
στ. $y = x_k$
ζ. $y > x_k$
Βρείτε το κάτω όριο των κλήσεων της διαδικασίας $TEST(i, j, i, k, y)$, χρησιμοποιώντας δέντρο απόφασης.

13. Έστω n διαστήματα $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, με $x_i, y_i \in \mathbb{R}, i \in 1, ..., n$. Σχεδιάστε και αναλύστε έναν αλγόριθμο εύρεσης του μεγαλύτερου πλήθους διαστημάτων με μη κενή τομή.

14. Έστω πίνακας ακεραίων A μέγέθους $n \times m$, όπου τα στοιχεία κάθε γραμμής και κάθε στήλης είναι ταξινομημένα με αύξουσα σειρά. Σχεδιάστε έναν αποφαντικό αλγόριθμο αναζήτησης στοιχείου x στον πίνακα A.

15. Ένας πίνακας μεγέθους $m \times n$ ονομάζεται $m \times n$ πίνακας $Young$ όταν τα στοιχεία κάθε γραμμής είναι ταξινομημένα από αριστερά προς τα δεξιά και τα στοιχεία κάθε στήλης είναι ταξινομημένα από πάνω προς τα κάτω. Κάποια από τα στοιχεία του πίνακα $Young$ μπορεί να έχουν τιμή ∞, και τα χειριζόμαστε ως μη υπαρκτά στοιχεία. Επομένως σε ένα πίνακα $Young$ μπορούν να αποθηκευτούν $r \leq mn$ πεπερασμένα στοιχεία.

α. Σχεδιάστε ένα 4×4 πίνακα $Young$, ο οποίος περιέχει τα στοιχεία $\{9, 16, 3, 2, 4, 8, 5, 14, 12\}$.

β. Δικαιολογείστε γιατί ένας $m \times n$ πίνακας $Young$ Y είναι άδειος εάν ισχύει $Y[1, 1] = \infty$. Δικαιολογείστε γιατί ο πίνακας Y είναι πλήρης (περιέχει mn στοιχεία) εάν ισχύει $Y[m, n] < \infty$.

γ. Σχεδιάστε έναν αλγόριθμο EXTRACT-MIN που επιστρέφει το ελάχιστο στοιχείο ενός μη κενού $m \times n$ πίνακα $Young$, με πολυπλοκότητα χρόνου $O(m+n)$. Ο αλγόριθμος θα πρέπει να χρησιμοποιεί μία αναδρομική υπορουτίνα η οποία επιλύει το πρόβλημα μεγέθους $m \times n$ επιλύοντας αναδρομικά είτε ένα υποπρόβλημα μεγέθους $(m-1) \times n$, είτε ένα υποπρόβλημα μεγέθους $m \times (n-1)$. Ορίστε ως $T(p)$, όπου $p = m+n$, το μέγιστο χρόνο εκτέλεσης του EXTRACT-MIN σε έναν οποιοδήποτε $m \times n$ πίνακα $Young$. Βρείτε μία αναδρομική σχέση για το χρόνο $T(p)$, από την οποία προκύπτει το κάτω όριο $O(m+n)$.

δ. Προτείνετε ένα τρόπο εισαγωγής ενός νέου στοιχείου σε ένα μη πλήρη $m \times n$ πίνακα $Young$, σε χρόνο $O(m+n)$.

ε. Χωρίς τη χρήση κάποιας υπορουτίνας ταξινόμησης, προτείνετε πώς μπορεί να χρησιμοποιηθεί ένας $m \times n$ πίνακας $Young$ για την ταξινόμηση n^2 στοιχείων σε $O(n^3)$ χρόνο.

στ. Σχεδιάστε έναν αλγόριθμο πολυπλοκότητας $O(m+n)$ για την αναζήτηση ενός στοιχείου σε ένα $m \times n$ πίνακα $Young$.

16. Κατά την ταξινόμηση μίας λίστας όπου δύο στοιχεία της μπορεί να είναι ίσα μεταξύ τους, η σύγκριση δύο στοιχείων μπορεί να έχει τρία πιθανά αποτελέσματα αντί για δύο. Σε αυτή την περιπτωση, οι αλγόριθμοι ταξινόμησης που βασίζονται στη σύγκριση αναπαρίστανται με εκτεταμένα τριαδικά δέντρα αναζήτησης αντί για δυαδικα, όπου κάθε εσωτερικός κόμβος $i : j$ έχει τρία παιδιά, που αντιστοιχούν στα τρία πιθανά αποτελέσματα.

Σχεδιάστε ένα εκτεταμένο τριαδικό δέντρο για την ταξινόμηση 4 στοιχείων, όπου όλα τα στοιχεία έχουν τιμή 0 ή 1. Συνεπώς όταν $K_1 < K_2$ και $K_3 < K_4$ ισχύει ότι $K_1 = K_3$ και $K_2 = K_4$.

Βρείτε τον ελάχιστο μέσο αριθμό συγκρίσεων που απαιτούνται, θεωρώντας ότι οι 2^4 δυνατές είσοδοι είναι ισοπίθανες.