Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
**Αλγόριθμοι και Πολυπλοκότητα**
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου
**1η Σειρά Γραπτών Ασκήσεων - Ημ/νία Παράδοσης 25/10/2018**

## Άσκηση 1: Ασυμπτωτικός Συμβολισμός, Αναδρομικές Σχέσεις.

(α) Να ταξινομήσετε τις παρακάτω συναρτήσεις σε αύξουσα σειρά τάξης μεγέθους, να βρείτε δηλαδή μια διάταξη $g_1, g_2, g_3, \ldots$ τέτοια ώστε $g_1 = O(g_2)$, $g_2 = O(g_3)$, κοκ. Σε αυτή τη διάταξη, να επισημάνετε τις συναρτήσεις που έχουν ίδια τάξη μεγέθους.

| | | | |
|---|---|---|---|
| $n^2$ | $2^{(\log_2 n)^4}$ | $\log(n!)/(\log n)^3$ | $n2^{2^{2^{100}}}$ |
| $\log(\binom{n}{\log n})$ | $(\log n)^2/\log\log n$ | $\log^4 n$ | $(\sqrt{n})!$ |
| $\binom{n}{6}$ | $n^3/(\log n)^8$ | $(\log_2 n)^{\log_2 n}$ | $\log\left(\binom{2n}{n}\right)$ |
| $n\sum_{k=0}^{n}\binom{n}{k}$ | $(\sqrt{n})^{\log_2\log_2(n!)}$ | $\sum_{k=1}^{n} k2^k$ | $\sum_{k=1}^{n} k2^{-k}$ |

(β) Να υπολογίσετε την τάξη μεγέθους $\Theta(\ )$ των λύσεων των παρακάτω αναδρομικών σχέσεων. Για όλες τις σχέσεις, να θεωρήσετε ότι $T(1) = \Theta(1)$.

1. $T(n) = 2T(n/3) + n\log n$
2. $T(n) = 3T(n/3) + n\log n$
3. $T(n) = 4T(n/3) + n\log n$
4. $T(n) = T(n/2) + T(n/3) + n$
5. $T(n) = T(n/2) + T(n/3) + T(n/6) + n$
6. $T(n) = T(n^{5/6}) + \Theta(\log n)$
7. $T(n) = T(n/4) + \sqrt{n}$.

## Άσκηση 2: Ταξινόμηση

(α) Θεωρούμε έναν πίνακα $A[1 \ldots n]$ με $n$ στοιχεία και τους υποπίνακες $A_1[1 \ldots \frac{n}{k}], A_2[\frac{n}{k}+1 \ldots 2\frac{n}{k}], \ldots, A_k[(k-1)\frac{n}{k}+1 \ldots n]$ που προκύπτουν από την διαμέριση του $A$ σε $k$ τμήματα με $n/k$ στοιχεία το καθένα (για απλότητα, υποθέτουμε ότι το $n$ είναι πολλαπλάσιο του $k$, μπορείτε ακόμη να υποθέσετε ότι τα $n$ και $k$ είναι δυνάμεις του 2). Θα λέμε ότι ο πίνακας $A$ είναι *ταξινομημένος κατά $k$ μέρη* όταν για κάθε $i$ και $j$, με $1 \leq i < j \leq k$, κάθε στοιχείο του υποπίνακα $A_i$ είναι μικρότερο ή ίσο από κάθε στοιχείο του υποπίνακα $A_j$. Δηλ. τα στοιχεία του $A$ έχουν ταξινομηθεί "εξωτερικά", μεταξύ των υποπινάκων, αλλά όχι κατ' ανάγκη και "εσωτερικά" στον ίδιο υποπίνακα. Π.χ., ο πίνακας $A = \big[[12, 14, 20, 18], [25, 22, 29, 32], [37, 42, 34, 50], [67, 59, 52, 76]\big]$ είναι ταξινομημένος κατά 4 μέρη.

1. Να διατυπώσετε *συγκριτικό* αλγόριθμο με χρόνο εκτέλεσης $O(n\log k)$ που ταξινομεί κατά $k$ μέρη έναν πίνακα $A$ με $n$ στοιχεία. Να δείξετε ακόμη ότι ο αλγόριθμός σας είναι βέλτιστος, δηλ. ότι κάθε συγκριτικός αλγόριθμος ταξινόμησης κατά $k$ μέρη ενός πίνακα με $n$ στοιχεία έχει χρόνο εκτέλεσης χειρότερης περίπτωσης $\Omega(n\log k)$.

2. Να διατυπώσετε συγκριτικό αλγόριθμο με χρόνο εκτέλεσης $O(n \log \frac{n}{k})$ που ταξινομεί πλήρως έναν πίνακα $A$ με $n$ στοιχεία που αρχικά είναι ταξινομημένος κατά $k$ μέρη. Να εξηγήσετε γιατί κάθε συγκριτικός αλγόριθμος για αυτό το πρόβλημα πρέπει να έχει χρόνο εκτέλεσης $\Omega(n \log \frac{n}{k})$.

(β) Έστω πίνακας ακεραίων $A[1 \dots n]$ που χαρακτηρίζεται από πολλαπλές εμφανίσεις των στοιχείων του. Συγκεκριμένα, θεωρούμε ότι το πλήθος των διαφορετικών στοιχείων του $A$ είναι μόλις πολυλογαριθμικό (δηλ. $\mathrm{O}(\log^d n)$, για κάποια σταθερά $d \geq 1$). Να διατυπώσετε έναν συγκριτικό αλγόριθμο που ταξινομεί τον πίνακα $A$ σε χρόνο $\mathrm{O}(n \log \log n)$. Γιατί δεν ισχύει το κάτω φράγμα του $\Omega(n \log n)$ σε αυτή την περίπτωση;

### Άσκηση 3: Διάστημα Ελάχιστου Μήκους που Καλύπτει όλους τους Πίνακες

(α) Δίνονται δύο ταξινομημένοι πίνακες ακεραίων $A_1[1 \dots n_1]$ και $A_2[1 \dots n_2]$. Θέλουμε να υπολογίσουμε θέσεις $i_1, i_2$ στους πίνακες $A_1, A_2$ ώστε να ελαχιστοποιηθεί η διαφορά $\big|A_1[i_1] - A_2[i_2]\big|$ (με άλλα λόγια, θέλουμε να υπολογίσουμε ένα διάστημα ελάχιστου μήκους που περιέχει τουλάχιστον ένα στοιχείο από κάθε πίνακα). Να διατυπώσετε αλγόριθμου *γραμμικού* χρόνου για αυτό το πρόβλημα. Να αιτιολογήσετε συνοπτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

(β) Δίνονται $m$ ταξινομημένοι πίνακες ακεραίων $A_1[1 \dots n_1]$, $A_2[1 \dots n_2]$, …, $A_m[1 \dots n_m]$. Αντίστοιχα με το (α), θέλουμε να υπολογίσουμε θέσεις $i_1, i_2, \dots, i_m$ στους πίνακες $A_1, A_2, \dots, A_m$ ώστε να ελαχιστοποιηθεί η διαφορά:

$$\max\big\{A_1[i_1], \dots, A_m[i_m]\big\} - \min\big\{A_1[i_1], \dots, A_m[i_m]\big\}$$

Να διατυπώσετε αλγόριθμο με χρόνο εκτέλεσης $O(mN)$ για αυτό το πρόβλημα, όπου $N = \sum_{k=1}^{m} n_k$. Να αιτιολογήσετε συνοπτικά την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

(γ) Ποια είναι λειτουργία που καθορίζει τον χρόνο εκτέλεσης του αλγόριθμου στο (β); Να εξηγήσετε πως (και κατά πόσο) μπορεί να επιταχυνθεί η υλοποίησή της. Με βάση αυτό, να βελτιώσετε τον χρόνο εκτέλεσης του αλγόριθμου.

### Άσκηση 4: Αναζήτηση

(α) Έχουμε 1.000.000 πανομοιότυπες φιάλες, μία από αυτές έχει ένα (άχρωμο, άοσμο, και άγευστο) πανίσχυρο μαγικό φίλτρο που προσφέρει απεριόριστη φυσική δύναμη και ευεξία, όλες οι άλλες έχουν καθαρό νερό. Το φίλτρο είναι αποτελεσματικό ακόμη και σε εξαιρετικά μικρές ποσότητες, όμως χρειάζεται 24 ώρες για να γίνουν εμφανή τα αποτελέσματά του. Να βρείτε τον ελάχιστο αριθμό εθελοντών (και τον τρόπο που θα τους αξιοποιήσουμε) που απαιτούνται για να εντοπίσουμε τη φιάλη με το μαγικό φίλτρο μέσα στις επόμενες 24 ώρες.

(β) Πρόκειται να πάμε ένα μεγάλο ταξίδι που θα διαρκέσει $k \geq 2$ ημέρες. Έχουμε επιλέξει τους $n > k$ σταθμούς του ταξιδιού (και τη σειρά με την οποία θα τους επισκεφθούμε) και έχουμε υπολογίσει τις αποστάσεις $d_1, d_2, \dots, d_n$ μεταξύ τους ($d_1$ είναι η απόσταση του πρώτου σταθμού από την αφετηρία, $d_2$ είναι η απόσταση του πρώτου από τον δεύτερο σταθμό, κοκ., όλες οι αποστάσεις είναι θετικοί ακέραιοι). Για να μην κουραστούμε, θέλουμε να προγραμματίσουμε το ταξίδι ώστε να ελαχιστοποιήσουμε τη μέγιστη απόσταση που θα διανύσουμε μέσα σε μία ημέρα (μπορούμε να σταματήσουμε για διανυκτέρευση μόνο σε κάποιον από τους επιλεγμένους σταθμούς). Να διατυπώσετε έναν αποδοτικό αλγόριθμο για αυτό το πρόβλημα. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

**Άσκηση 5: Επιλογή**

(α) Έστω πολυσύνολο (multiset) $S$ με $n$ θετικούς ακέραιους που όλοι είναι μικρότεροι ή ίσοι δεδομένου ακεραίου $M$. Έχουμε πρόσβαση (μόνο) στην κατανομή $F_S$ των στοιχείων της συλλογής. Συγκεκριμένα, έχουμε στη διάθεσή μας συνάρτηση $F_S(\ell)$ που για κάθε φυσικό $\ell$, επιστρέφει το πλήθος των στοιχείων του $S$ που δεν ξεπερνούν το $\ell$, δηλ. $F_S(\ell) = \big|\{x \in S : x \leq \ell\}\big|$. Να διατυπώσετε αποδοτικό αλγόριθμο που δέχεται ως είσοδο φυσικό $k$, $1 \leq k \leq n$, και υπολογίζει (καλώντας την $F_S$) το $k$-οστό μικρότερο στοιχείο του $S$. Να αιτιολογήσετε την ορθότητα του αλγορίθμου σας και να προσδιορίσετε το πλήθος των απαιτούμενων κλήσεων στην $F_S$ (στη χειρότερη περίπτωση). Προσπαθήστε το πλήθος των κλήσεων στην $F_S$ να μην εξαρτάται από το $n$ (μπορεί όμως να εξαρτάται από το $M$).

(β) Έστω πίνακας διαφορετικών θετικών ακεραίων $A[1 \ldots n]$ και έστω $M$ το μέγιστο στοιχείο του $A$. Θεωρούμε το πολυσύνολο $S$ που αποτελείται από όλες τις μη αρνητικές διαφορές ζευγών στοιχείων του $A$. Δηλαδή, έχουμε ότι:

$$S = \big\{A[i] - A[j] : i \neq j \text{ και } A[i] > A[j]\big\}$$

Να διατυπώσετε αποδοτικό αλγόριθμο που υπολογίζει το $k$-οστό μικρότερο στοιχείο του $S$. Να προσδιορίσετε την υπολογιστική πολυπλοκότητα του αλγορίθμου σας (συναρτήσει των $n$ και $M$) και να αιτιολογήσετε την ορθότητά του. *Υπόδειξη:* Προσπαθήστε να υλοποιήσετε αποδοτικά την $F_S$ και να χρησιμοποιήσετε τον αλγόριθμο του (α).