Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου, Θανάσης Λιανέας
1η Σειρά Γραπτών Ασκήσεων - Ημ/νία Παράδοσης 5/11/2020

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 $g_1 = \mathrm{O}(g_2)$, $g_2 = \mathrm{O}(g_3)$, κοκ. Σε αυτή τη διάταξη, να επισημάνετε τις συναρτήσεις που έχουν ίδια τάξη μεγέθους.

$(\sqrt{n})^{\log_2 \log_2 (n!)}$	$2^{(\log_2 \log_2 n)^4}$	$\log(n!)/(\log\log n)^5$	$n2^{2^{2^{100}}}$
$n \sum_{k=0}^{n} \binom{n}{k}$	$\log\left(\binom{2n}{n}\right)$	$\sum_{k=1}^{n} k2^k$	$\sum_{k=1}^{n} k2^{-k}$

(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. $T(n) = 6\,T(n/3) + n^2 \log n$
2. $T(n) = 9\,T(n/3) + n^2 \log n$
3. $T(n) = 11\,T(n/3) + n^2 \log n$
4. $T(n) = T(n/4) + T(n/2) + n$
5. $T(n) = 2\,T(n/4) + T(n/2) + n$
6. $T(n) = T(n^{2/3}) + \Theta(\log n)$
7. $T(n) = T(n/3) + \sqrt{n}$.

Άσκηση 2: Προθεματική Ταξινόμηση

Έστω μη ταξινομημένος πίνακας $A[1..n]$ αρχικοποιημένος σε μια μετάθεση του συνόλου $\{1, \ldots, n\}$. Θέλουμε να ταξινομήσουμε τον πίνακα A χρησιμοποιώντας **μόνο** *προθεματικές περιστροφές*. Μια προθεματική περιστροφή υλοποιείται επιλέγοντας κάποιο $k \leq n$ και αντιστρέφοντας τη σειρά των k αριστερότερων στοιχείων του πίνακα, ως εξής:

$$\Big[\underline{A[1], A[2], \ldots, A[k-1], A[k]}, A[k{+}1], \ldots, A[n]\Big] \;\rightarrow\; \Big[\underline{A[k], A[k-1], \ldots, A[2], A[1]}, A[k{+}1], \ldots, A[n]\Big]$$

Στόχος μας είναι να ταξινομήσουμε τον πίνακα A (σε αύξουσα σειρά) κάνοντας όσο το δυνατόν λιγότερες προθεματικές περιστροφές στην χειρότερη περίπτωση.

(α) Να διατυπώσετε αλγόριθμο που ταξινομεί τον A χρησιμοποιώντας το πολύ $2n$ προθεματικές περιστροφές. Εδώ η ανάλυσή σας οφείλει να είναι κατά το δυνατόν ακριβής και να προσδιορίζει πλήρως και τις σταθερές (π.χ., το $2n - 3$ θεωρείται καλύτερο από το $2n$).

(β) Θεωρούμε τώρα ότι οι επιτρεπτές μας κινήσεις είναι οι *προσημασμένες προθεματικές περιστροφές*. Δηλαδή, κάθε φορά που ένα στοιχείο του τρέχοντος πίνακα συμμετέχει σε μία περιστροφή, αντιστρέφουμε το πρόσημό του:

$$\Big[\underline{A[1], A[2], \ldots, A[k-1], A[k]}, A[k+1], \ldots, A[n]\Big] \;\rightarrow$$
$$\Big[\underline{-A[k], -A[k-1], \ldots, -A[2], -A[1]}, A[k+1], \ldots, A[n]\Big]$$

Ο στόχος μας παραμένει να ταξινομήσουμε τα στοιχεία έτσι ώστε να εμφανίζονται τελικά σε αύξουσα σειρά και με τα αρχικά (θετικά) τους πρόσημα, παρόλο που στα ενδιάμεσα βήματα, κάποια πρόσημα μπορεί (προσωρινά) να γίνουν αρνητικά.

Να διατυπώσετε αλγόριθμο που ταξινομεί τον πίνακα A χρησιμοποιώντας το πολύ $3n$ προσημασμένες προθεματικές περιστροφές.

(γ) Θα λέμε ότι ένα ζεύγος στοιχείων $(A_t[i], A_t[i+1])$ σε έναν ενδιάμεσο πίνακα A_t (που προκύπτει από τον $A = A_0$ μετά από t διαδοχικές προσημασμένες προθεματικές περιστροφές) είναι *συμβατό* όταν τα στοιχεία $A_t[i]$ και $A_t[i+1]$ έχουν το ίδιο πρόσημο, η απόλυτη τιμή των στοιχείων $A_t[i]$ και $A_t[i+1]$ διαφέρει κατά 1 (άρα οι φυσικοί αριθμοί $|A_t[i]|$ και $|A_t[i+1]|$ είναι διαδοχικοί) και τα στοιχεία $A_t[i]$ και $A_t[i+1]$ είναι ταξινομημένα στον πίνακα A_t (λαμβάνοντας υπόψη και το πρόσημό τους).

Για παράδειγμα στον πίνακα $[4, -2, -1, 3]$, το ζευγάρι $(-2, -1)$ είναι συμβατό (και κανένα άλλο), γιατί $-2 < -1$ και τα $|-2| = 2$ και $|-1| = 1$ είναι διαδοχικοί φυσικοί αριθμοί.

1. Να δείξετε ότι για κάθε ενδιάμεσο πίνακα A_t που είναι διαφορετικός από τον $[-1, -2, \ldots, -n]$, μπορούμε να κατασκευάσουμε ένα νέο συμβατό ζεύγος έπειτα από 2 το πολύ προσημασμένες προθεματικές περιστροφές. Να θεωρήσετε, καταχρηστικά, τη μετακίνηση του μεγαλύτερου στοιχείου του A_t στην τελευταία θέση, με θετικό πρόσημο, ως δημιουργία συμβατού ζεύγους.
2. Να διατυπώσετε αλγόριθμο που ταξινομεί τον A χρησιμοποιώντας το πολύ $2n$ προσημασμένες προθεματικές περιστροφές.

Άσκηση 3: Παίζοντας Χαρτιά

Έστω μια τυχαία ανακατεμένη τράπουλα με n κάρτες, αριθμημένες από 1 μέχρι και n. Τραβάμε συνεχώς φύλλα από την κορυφή της τράπουλας (ενόσω υπάρχουν) και τα τοποθετούμε σύμφωνα με τους παρακάτω κανόνες:

– Η πρώτη κάρτα μπαίνει στην κορυφή μιας στοίβας.
– Για κάθε επόμενη κάρτα c, είτε (i) η c τοποθετείται στην κορυφή μιας υπάρχουσας στοίβας, για την οποία η αξία της κάρτας c' στην κορυφή της είναι μεγαλύτερη από την αξία της c (οπότε η c τοποθείται πάνω από την c' και γίνεται κορυφή της στοίβας), είτε (ii) δημιουργείται μια νέα στοίβα δεξιότερα από τις υπάρχουσες με μοναδικό στοιχείο (και κορυφή) τη c.

Το παιχνίδι ολοκληρώνεται σε n γύρους, όταν έχουν εξαντληθεί οι κάρτες από την τράπουλα.

Παράδειγμα. Θεωρούμε μια τράπουλα με 6 φύλλα στην εξής σειρά: $6, 3, 5, 2, 4, 1$. Στο 1ο γύρο, το 6 δημιουργεί μια στοίβα. Στον 2ο γύρο, το 3 τοποθετείται πάνω από το 6 στην πρώτη στοίβα. Στον 3ο γύρο, το 5 είναι μεγαλύτερο του 3 και έτσι πρέπει να ξεκινήσει καινούργια στοίβα (οπότε έχουμε την κατάσταση $[(6, 3), (5)]$). Στον 4ο γύρο, το 2 μπορεί να τοποθετηθεί στην κορυφή οποιασδήποτε από τις δύο στοίβες (οπότε έχουμε είτε την κατάσταση $[(6, 3, 2), (5)]$ είτε την κατάσταση $[(6, 3), (5, 2)]$). Ανάλογα με την επιλογή μας στον 4ο γύρο, το παιχνίδι μπορεί να καταλήξει είτε στην κατάσταση $[(6, 3, 2, 1), (5, 4)]$ είτε σε μία από τις καταστάσεις $[(6, 3, 1), (5, 2), (4)]$ ή $[(6, 3), (5, 2, 1), (4)]$ ή $[(6, 3), (5, 2), (4, 1)]$.

1. Να διατυπώσετε αποδοτικό αλγόριθμο που ελάχιστοποιεί το πλήθος των στοιβών που δημιουργούνται κατά τη διάρκεια του παιχνιδιού. Να αιτιολογήσετε την ορθότητα του αλγορίθμου σας και να εκτιμήσετε (και να προσπαθήσετε να βελτιστοποιήσετε) την υπολογιστική πολυπλοκότητα του αλγορίθμου σας στη χειρότερη περίπτωση.
2. Εξηγήστε πως μπορούμε να εφαρμόσουμε τον αλγόριθμο του (1) για να ταξινομήσουμε (με χρονικά αποδοτικό τρόπο) την τράπουλα. Ποιος είναι ο χρόνος εκτέλεσης του αλγόριθμου ταξινόμησης που προκύπτει στη χειρότερη περίπτωση;

3. Να περιγράψετε τη λειτουργία των αλγόριθμων που διατυπώσατε στα (1) και (2) με είσοδους $3, 2, 4, 7, 8, 1, 5, 6$. Πόσες στοίβες προκύπτουν; Ποιο είναι το μήκος της μέγιστης αύξουσας υπακολουθίας της εισόδου;
4. Να δείξετε ότι το πλήθος των στοιβών στις οποίες καταλήγουμε (ανεξάρτητα από τον αλγόριθμο που ακολουθούμε) είναι πάντα μεγαλύτερο ή ίσο του μήκους της μέγιστης αύξουσας υπακολουθίας της εισόδου.
5. **Bonus:** Σε σχέση με το πρόβλημα της μέγιστης αύξουσας υπακολουθίας, τι ρόλο παίζει το κορυφαίο στοιχείο της k-οστής (μετρώντας από τα αριστέρα προς τα δεξιά) στοίβας; Με βάση αυτό, να εξηγήσετε γιατί ο αλγόριθμος του (1) λύνει το πρόβλημα της μέγιστης αύξουσας υπακολουθίας. Τι παρατηρείτε σε σχέση με το μήκος της μέγιστης φθίνουσας υπακολουθίας;
6. Να δείξετε ότι για κάθε ανακάτεμα μιας τράπουλας με $nm + 1$ φύλλα, ο αλγόριθμος του (1) δημιουργεί είτε τουλάχιστον $n + 1$ στοίβες είτε μια στοίβα μεγέθους τουλάχιστον $m + 1$. Μπορείτε να συνδέσετε αυτή την παρατήρησή σας με γνωστό θεώρημα των Διακριτών Μαθηματικών; Να δώσετε μια ακολουθία εισόδου με 25 κάρτες όπου ο αλγόριθμος του (1) δημιουργεί 5 στοίβες μεγέθους 5.

Άσκηση 4: Γρήγορη Επιλογή στο Πεδίο της Μάχης

Εδώ και κάποια χρόνια, οι κάτοικοι της χώρας των Αλγορίθμων δεν μπορούν να συμφωνήσουν ως προς το πόσο σημαντικοί είναι οι αλγόριθμοι της Δυαδικής Αναζήτησης και της Quickselect! Οι κάτοικοι έχουν χωριστεί σε δύο στρατόπεδα και κατέχουν δύο εκτοξευτές σωματιδίων τους οποίους θα χρησιμοποιήσουν σαν όπλο. Θεωρούμε πως το πεδίο μάχης έχει τη μορφή μιας απέραντης ευθείας. Οι δύο αντιμαχόμενες πλευρές τοποθετούν τους εκτοξευτές τους στις θέσεις $x = 0$ και $x = L$. Ο εκτοξευτής του πρώτου στρατοπέδου εκπέμπει σωματίδια τύπου a προς τα δεξιά και ο εκτοξευτής του δεύτερου στρατόπεδου εκπέμπει σωματίδια τύπου b προς τα αριστερά. Θεωρούμε πως η μάχη ξεκινάει τη χρονική στιγμή $t = 0$, και ο τρόπος με τον οποίο αλληλεπιδρούν τα σωματίδια καθορίζεται από τους παρακάτω κανόνες:

- Αν δύο σωματίδια διαφορετικού τύπου συγκρουστούν μεταξύ τους, τότε αυτά εξαϋλωνονται.
- Αν δύο σωματίδια ίδιου τύπου συγκρουστούν μεταξύ τους, τότε το ένα προσπερνάει το άλλο χωρίς να εξαϋλωθούν.

Η τεχνολογία των εκτοξευτών επιτρέπει στην ταχύτητα των εκπεμπόμενων σωματιδίων να μεταβάλεται διαρκώς. Έτσι, τα εκπεμπόμενα σωματίδια πραγματοποιούν κίνηση με σταθερή κατεύθυνση (δηλ. τα σωματίδια τύπου a κινούνται πάντα προς τα δεξιά και τα σωματίδια τύπου b κινούνται πάντα προς τα αριστερά), αλλά όχι απαραίτητα ομαλή (δηλ. η ταχύτητα ενός σωματιδίου μπορεί να αυξομειώνεται αυθαίρετα με τον χρόνο). Γνωρίζουμε μόνο ότι κάθε αντιμαχόμενη πλευρά εκτοξεύει n σωματίδια και ότι η ταχύτητα κάθε σωματιδίου δεν μπορεί ποτέ να ξεπεράσει το $V_{\max}$ και δεν μπορεί ποτέ να είναι μικρότερη από $V_{\min} > 0$.

Οι στρατηγοί των δύο στρατοπέδων ενδιαφέρονται να μάθουν πότε γίνεται η πρώτη σύγκρουση σωματιδίων και ποια είναι τα σωματίδια που συμμετέχουν σε αυτή. Να διατυπώσετε έναν αποδοτικό αλγόριθμο για αυτό το πρόβλημα και να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας, σε καθεμία από τις παρακάτω περιπτώσεις:

- Η πλευρά που εκτοξεύει τα σωματίδια a έχει στη διάθεσή της έναν υπερυπολογιστή που για κάθε σωματίδιο i (το i μπορεί να είναι τύπου a ή τύπου b) και για κάθε χρονική στιγμή $t \geq 0$ υπολογίζει σε σταθερό χρόνο (και με απόλυτη ακρίβεια) τη θέση του i τη χρονική στιγμή t, αν το i είναι το μοναδικό σωματίδιο στο πεδίο της μάχης.

– Η πλευρά που εκτοξεύει τα σωματίδια b έχει στη διάθεσή της έναν άλλο υπερυπολογιστή που για κάθε ζευγάρι σωματιδίων i (τύπου a) και j (τύπου b) υπολογίζει σε σταθερό χρόνο (και με απόλυτη ακρίβεια) τη χρονική στιγμή t που τα δύο σωματίδια συγκρούονται.

Στην πρώτη (αντ. δεύτερη) περίπτωση, ο υπερυπολογιστής λαμβάνει ως είσοδο ένα ζεύγος (i, t) (αντ. (i, j)) και επιστρέψει τη θέση του σωματιδίου i τη χρονική στιγμή t (αντ. τη χρονική στιγμή που τα σωματίδια i και j συγκρούονται) και μόνο αυτό. Σε καθεμία από τις δύο περιπτώσεις, ο αλγόριθμός σας πρέπει να ελαχιστοποιεί το πλήθος των κλήσεων στον υπερυπολογιστή (ως συνάρτηση των n, L, $V_{\max}$ και $V_{\min}$).

Άσκηση 5 - Bonus: Ερωτήματα Ανήκειν

Βασιζόμενοι σε hashing, θέλουμε να αναπτύξουμε μια δομή δεδομένων που διατηρεί μια σύνοψη ενός συνόλου S, αποτελούμενου από n θετικούς φυσικούς αριθμούς, και υλοποιεί αποδοτικά (πρακτικά, σε σταθερό χρόνο) τις παρακάτω δύο λειτουργίες: "πρόσθεσε το x στο S" και "έλεγξε αν $x \in S$". Για αυτό τον σκοπό, διατηρούμε μια σύνοψη του S σε πίνακα A με m θέσεις, μεγέθους ενός δυαδικού ψηφίου η καθεμία, και χρησιμοποιούμε συναρτήσεις hash $h : \mathbb{N}_+ \to [m]$ τέτοιες ώστε $\mathbb{P}\text{rob}[h(x) = j] = 1/m$, για κάθε $x \in \mathbb{N}_+$ και για κάθε $j \in [m]$. Απαιτούμε, τέλος, η χρήση τυχαιότητας να μην οδηγεί σε false negatives (δηλ. κάθε αρνητική απάντηση στο ερώτημα "$x \in S$?" πρέπει να είναι σωστή), ενώ μπορεί να οδηγεί σε false positives (δηλ. μπορεί μια θετική απάντηση στο ερώτημα "$x \in S$?" να είναι λανθασμένη) με μικρή όμως πιθανότητα.

(α) Σχεδιάστε μια τέτοια δομή δεδομένων με χρήση μιας μόνο hash συνάρτησης και υπολογίστε την πιθανότητα μιας false positive απάντησης. Ποια είναι η πιθανότητα για false positive απάντηση αν $m = 8n$;

(β) Σχεδιάστε μια τέτοια δομή δεδομένων με χρήση $k \geq 1$ ανεξάρτητων hash συναρτήσεων και υπολογίστε την πιθανότητα μιας false positive απάντησης. Ποια είναι η βέλτιστη τιμή του k και ποια η αντίστοιχη πιθανότητα για false positive απάντηση αν $m = 8n$;

Άσκηση 6 - Bonus: Προθέματα Συμβολοσειρών

Δίνεται ένα σύνολο S αποτελούμενο από n συμβολοσειρές μήκους $\ell_1, \dots, \ell_n$. Συμβολίζουμε με $N = \ell_1 + \cdots + \ell_n$ το συνολικό μήκος των συμβολοσειρών στο σύνολο S. Θέλουμε να αποθηκεύσουμε τις συμβολοσειρές του S ώστε να μπορούμε να απαντάμε αποδοτικά ερωτήματα μέγιστου κοινού προθέματος της μορφής: "Δίνονται k συμβολοσειρές $s_1, \dots, s_k \in S$. Ποιο είναι το μέγιστο κοινό πρόθεμά τους;" Ποια δομή δεδομένων θα χρησιμοποιήσετε για την αποδοτική αποθήκευση του S και ποια δομή δεδομένων / λειτουργία θα χρησιμοποιήσετε (και πως) για την αποδοτική απάντηση τέτοιων ερωτημάτων μέγιστου κοινού προθέματος; Ποιες είναι οι απαιτήσεις σε χρόνο και μνήμη για την κατασκευή της δομής αποθήκευσης του S, και ποια είναι η χρονική πολυπλοκότητα για την δημιουργία της αντίστοιχης δομής και την απάντηση ερωτημάτων μέγιστου κοινού προθέματος για k συμβολοσειρές του S; (Οι απαντήσεις σας εδώ ας είναι συνοπτικές - αρκεί μια απλή αναφορά στις δομές δεδομένων που θα χρησιμοποιήσετε, στις απαιτήσεις σε χώρο και χρόνο, και στον τρόπο με τον οποίο θα απαντηθούν τα ερωτήματα μέγιστου κοινού προθέματος).