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 4/11/2019

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.

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

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

Άσκηση 2: Διάστημα Ελάχιστου Μήκους που Καλύπτει όλους τους Πίνακες

(α) Δίνονται δύο ταξινομημένοι πίνακες ακεραίων $A_1[1 \ldots n_1]$ και $A_2[1 \ldots n_2]$. Θέλουμε να υπολογίσουμε θέσεις i_1, i_2 στους πίνακες A_1, A_2 ώστε να ελαχιστοποιηθεί η διαφορά $\left|A_1[i_1] - A_2[i_2]\right|$ (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 $A_1[1 \ldots n_1]$, $A_2[1 \ldots n_2]$, $\ldots$, $A_m[1 \ldots n_m]$. Αντίστοιχα με το (α), θέλουμε να υπολογίσουμε θέσεις $i_1, i_2, \ldots, i_m$ στους πίνακες $A_1, A_2, \ldots, A_m$ ώστε να ελαχιστοποιηθεί η διαφορά:

$$\max\left\{A_1[i_1], \ldots, A_m[i_m]\right\} - \min\left\{A_1[i_1], \ldots, A_m[i_m]\right\}$$

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 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

Τα προηγούμενα δύο χρόνια, οι κάτοικοι της χώρας των Αλγορίθμων είχαν χωριστεί σε δύο στρατόπεδα, διαφωνώντας για το πόσο σημαντική είναι η Δυαδική Αναζήτηση (για λεπτομέρειες, δείτε την

εκφώνηση της 1ης προγραμματιστικής άσκησης των δύο προηγούμενων ετών). Η ειρήνη δεν φαίνεται ότι θα κρατήσει για πολύ, αφού έχουν ξεσπάσει νέες διαφωνίες για πόσο σημαντικός αλγόριθμος είναι η Quickselect!

Όπως και στο παρελθόν, οι κάτοικοι έχουν χωριστεί σε δύο στρατόπεδα και κατέχουν δύο (πολύ προηγμένους πλέον) 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 έχει προοδεύσει τόσο πολύ που 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 σωματίδιου 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.
- 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.

Άσκηση 5: Κρυμμένος Θησαυρός

Όταν οι εισβολείς της φυλής Heuristics αποχώρησαν νικημένοι από τη χώρα των Αλγορίθμων, δεν κατάφεραν να πάρουν μαζί τους κανένα λάφυρο. Τα έθαψαν όλα σε άγνωστη θέση x κατά μήκος της κεντρικής αρτηρίας της χώρας. Ο Πρόεδρος της χώρας σας ανέθεσε να φέρετε στο φως αυτόν τον σημαντικό θησαυρό. Για τον σκοπό αυτό, έχετε προμηθευτεί έναν ανιχνευτή μετάλλων που μπορεί να ανιχνεύσει τον θησαυρό, αν βρεθεί έστω και στιγμιαία πάνω από αυτόν.

Θεωρείτε ότι η κεντρική οδική αρτηρία της χώρας αντιστοιχεί στην ευθεία των ακεραίων αριθμών, και θα ξεκινήσετε την αναζήτησή σας από τη θέση 0. Η θέση x του κρυμμένου θησαυρού είναι άγνωστη και μπορεί να είναι οποιοσδήποτε αριθμός του $\mathbb{Z}$. Το μόνο που μπορείτε να κάνετε είναι να κινείστε, πέρα - δώθε, στην κεντρική αρτηρία, μέχρι να περάσετε πάνω από τη θέση x και ο ανιχνευτής σας να εντοπίσει τον θησαυρό. Για τον προγραμματισμό των κινήσεών σας, θέλετε να χρησιμοποιήσετε έναν αποδοτικό αλγόριθμο που θα σας επιτρέψει να ανακαλύψετε τη θέση x διανύοντας απόσταση

$O(|x|)$. Να διατυπώσετε έναν τέτοιο αλγόριθμο και να δείξετε ότι υπάρχει σταθερά c (για την οποία πρέπει να υπολογίσετε ένα, κατά το δυνατόν ακριβές, άνω φράγμα) έτσι ώστε ο αλγόριθμος σας να εξασφαλίζει ότι (στη χειρότερη περίπτωση) θα βρείτε τον θησαυρό μετά από $c|x|$ βήματα, το πολύ.