Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Αλγόριθμοι και Πολυπλοκότητα

Μεταπτυχιακό ΜΠΛΑ, ΣΕΜΦΕ

Διδάσκοντες: Σ. Ζάχος, Δ. Φωτάκης

Προτεινόμενες Ασκήσεις

Άσκηση 1. Σε μια φανταστική χώρα, διενεργείται δημοσκόπηση σχετικά με τη στάση των πολιτών ως προς μια σημαντική πολιτικο-οικονομική μεταβολή. Οι πολίτες απαντούν στη δημοσκόπηση με ‘‘ναι’’ ή ‘‘όχι’’ (υπέρ ή εναντίον της μεταβολής). Αν το (πραγματικό) ποσοστό των πολιτών που τάσσονται υπέρ της μεταβολής είναι p, θέλουμε να υπολογίσουμε μια εκτίμηση $\hat{p}$ του p ώστε $\Pr[|\hat{p} - p| \leq \varepsilon p] > 1 - \delta$, για δεδομένα $\varepsilon, \delta \in (0, 1)$. Για τη δημοσκόπηση, θα ρωτήσουμε N πολίτες, που επιλέγονται ισοπίθανα και ανεξάρτητα από το σύνολο των πολιτών. Η εκτίμησή μας $\hat{p}$ θα είναι το ποσοστό των N πολιτών που τάσσονται υπέρ της μεταβολής. Χρησιμοποιώντας Chernoff bounds, να υπολογίσετε (ως συνάρτηση των ε, δ, και p) το ελάχιστο μέγεθος N του δείγματος που χρειαζόμαστε. Βρείτε την τιμή του N για $\varepsilon = 0.02$ και $\delta = 0.05$, αν γνωρίζουμε ότι $p \in [0.2, 0.8]$ (και δείτε ότι αυτή η τιμή είναι ανεξάρτητη του πληθυσμού της χώρας!).

Άσκηση 2. Θεωρούμε ένα ασύρματο δίκτυο όπου ένας δέκτης ‘‘ακούει’’ μια πηγή μετάδοσης σε μια συγκεκριμένη συχνότητα, και το bit $b(t)$, που μεταδίδεται τη χρονική στιγμή t, αναπαρίσταται είτε ως 1 είτε ως -1. Εκτός όμως από την πηγή μετάδοσης με την οποία είναι συντονισμένος ο δέκτης μας, υπάρχουν και άλλες n πηγές στον περιβάλλοντα χώρο, οι οποίες δημιουργούν παρεμβολές. Θεωρούμε ότι κάθε τέτοια πηγή i έχει ισχύ $p_i \in (0, 1)$, και μεταδίδει το bit $b_i(t) \in \{1, -1\}$ τη χρονική στιγμή t. Ας δεχθούμε ότι ο δέκτης μας λαμβάνει το σήμα:

$$s(t) = b(t) + \sum_{i=1}^{n} p_i b_i(t)$$

και συμπεραίνει ότι η πηγή μετέδωσε 1, αν $s(t) \geq 0$, και -1, διαφορετικά. Αν θεωρήσουμε ότι τα bits $b_i(t)$ επιλέγονται ομοιόμορφα και ανεξάρτητα, να εκτιμήσετε (με χρήση Chernoff bounds) την πιθανότητα ο δέκτης μας να υπολογίσει λάθος το $b(t)$.

Άσκηση 3. Ένας πίνακας ακεραίων $A[1 \ldots n]$ είναι *σχεδόν ταξινομημένος* αν υπάρχουν $n/4$ το πολύ στοιχεία τα οποία αν διαγράψουμε, ο υποπίνακας του A που απομένει είναι ταξινομημένος. Για παράδειγμα, ο πίνακας $[1, 2, 3, 5, 4, 7, 9, 6, 10, 11, 12, 8]$ είναι σχεδόν ταξινομημένος, αφού η διαγραφή των στοιχείων 5, 6, και 8 δίνει τον πίνακα $[1, 2, 3, 4, 7, 9, 10, 11, 12]$, που είναι ταξινομημένος. Για απλότητα, υποθέτουμε στη συνέχεια ότι όλα τα στοιχεία του πίνακα A είναι διαφορετικά.

Στοχεύουμε στη διατύπωση ενός πιθανοτικού αλγόριθμου *λογαριθμικού χρόνου* που θα διακρίνει πίνακες που είναι (πλήρως) ταξινομημένοι από πίνακες που δεν είναι σχεδόν ταξινομημένοι[1]. Συγκεκριμένα, αν ο πίνακας A είναι πλήρως ταξινομημένος, ο αλγόριθμος θα αποφαίνεται πάντα (δηλ. με πιθανότητα 1) ότι ο A είναι σχεδόν ταξινομημένος. Αν ο πίνακας A δεν είναι σχεδόν ταξινομημένος, ο αλγόριθμος θα αποφαίνεται, με πιθανότητα τουλάχιστον 0.9, ότι ο A δεν είναι σχεδόν ταξινομημένος.

[1] Εκ πρώτης όψης, αυτό είναι ένας φιλόδοξος στόχος. Δείτε ότι οποιοσδήποτε ντετερμινιστικός αλγόριθμος για αυτό το πρόβλημα πρέπει να έχει τουλάχιστον γραμμικό χρόνο εκτέλεσης, αφού θα πρέπει να διαβάσει τουλάχιστον $3n/4$ από τα στοιχεία του πίνακα A.

(α) Θεωρούμε τον αλγόριθμο που επιλέγει τυχαία (και ανεξάρτητα) k θέσεις $a_1, \dots, a_k$ του πίνακα A, και αποφαίνεται ότι ο πίνακας A είναι σχεδόν ταξινομημένος αν για κάθε θέση a_i που έχει επιλεγεί, ισχύει ότι $A[a_i - 1] \le A[a_i] \le A[a_i + 1]$. Αν υπάρχει έστω μία θέση a_i που δεν ικανοποιεί αυτό το κριτήριο, ο αλγόριθμος αποφαίνεται ότι ο πίνακας A δεν είναι σχεδόν ταξινομημένος. Να δώσετε παράδειγμα πίνακα A με n στοιχεία όπου ο αλγόριθμος χρειάζεται να ελέγξει $k = \Omega(n)$ θέσεις ώστε η πιθανότητα λάθους να γίνει μικρότερη του 0.1.

(β) Υποθέτουμε ότι εφαρμόζουμε την παρακάτω εκδοχή της Δυαδικής Αναζήτησης σε έναν πίνακα A που μπορεί να μην είναι ταξινομημένος (με κίνδυνο φυσικά η αναζήτηση να αποτύχει, αν και το στοιχείο x υπάρχει στον A):

> BINARY-SEARCH$(A, x, \text{low}, \text{up})$
>   **if** low = up **then return** low;
>   **else** mid $\leftarrow \lceil (\text{low} + \text{up})/2 \rceil$;
>   **if** $x < A[\text{mid}]$ **then return** BINARY-SEARCH$(A, x, \text{low}, \text{mid} - 1)$;
>   **else return** BINARY-SEARCH$(A, x, \text{mid}, \text{up})$;

Έστω λοιπόν ότι για κάποιες τιμές x, y, η BINARY-SEARCH$(A, x, 1, n)$ επιστρέφει τη θέση k και η BINARY-SEARCH$(A, y, 1, n)$ επιστρέφει τη θέση ℓ. Να δείξετε ότι αν $k < \ell$, τότε $x < y$.

(γ) Θεωρούμε τώρα τον αλγόριθμο που επιλέγει τυχαία (και ανεξάρτητα) k θέσεις $a_1, \dots, a_k$ του πίνακα A, και αποφαίνεται ότι ο πίνακας A είναι σχεδόν ταξινομημένος αν για κάθε θέση a_i που έχει επιλεγεί, ισχύει ότι $a_i =$ BINARY-SEARCH$(A, A[a_i], 1, n)$. Αν υπάρχει έστω μία θέση a_i που δεν ικανοποιεί αυτό το κριτήριο, ο αλγόριθμος αποφαίνεται ότι ο πίνακας A δεν είναι σχεδόν ταξινομημένος. Χρησιμοποιώντας το (β), να δείξετε ότι για κάθε πίνακα A με n θέσεις, αρκεί ο αλγόριθμος να ελέγξει $k = \Theta(1)$ θέσεις, ώστε η πιθανότητα λάθους να γίνει μικρότερη του 0.1.

Άσκηση 4. Στο πρόβλημα του MAX k-CUT, δίνεται ένα απλό μη κατευθυνόμενο γράφημα $G(V, E)$ με θετικά βάρη στις ακμές. Το ζητούμενο είναι μια διαμέριση του V σε k σύνολα $V_1, \dots, V_k$ που μεγιστοποιεί το συνολικό βάρος των ακμών στο αντίστοιχο k-cut. Θεωρούμε τον απλό πιθανοτικό αλγόριθμο που ισοπίθανα τοποθετεί κάθε κορυφή σε ένα από τα σύνολα $V_1, \dots, V_k$. Να δείξετε ότι η μέση τιμή του συνολικού βάρους των ακμών στο k-cut που προκύπτει είναι τουλάχιστον $(1 - \frac{1}{k})$ OPT.

Άσκηση 5. Να δείξετε ότι ο άπληστος αλγόριθμος για το (διακριτό) Knapsack που χρησιμοποιεί ως κριτήριο επιλογής το λόγο της αξίας προς το μέγεθος κάθε αντικειμένου δεν επιτυγχάνει σταθερό λόγο προσέγγισης. Να τροποποιήσετε τον αλγόριθμο ώστε (η τροποποιημένη εκδοχή) να επιτυγχάνει λόγο προσέγγισης 2 (ο χρόνος εκτέλεσης πρέπει να παραμείνει $O(n \log n)$, όπου n ο αριθμός των αντικειμένων). Στη συνέχεια, να γενικεύσετε την προσέγγισή σας ώστε για κάθε σταθερά $k \ge 1$, ο αλγόριθμος να επιτυγχάνει λόγο προσέγγισης $(k+1)/k$ σε χρόνο $O(n^{k+1})$.

Άσκηση 6. Στο πρόβλημα του Maximum Coverage, δίνονται ένα σύνολο U με n στοιχεία, μια συλλογή $S_1, \dots, S_m$ από υποσύνολα του U, και ένας θετικός ακέραιος k. Το ζητούμενο είναι η επιλογή k υποσυνόλων $S_{i_1}, \dots, S_{i_k}$ ώστε να μεγιστοποιηθεί ο αριθμός των στοιχείων που καλύπτονται από αυτά (δηλ. να μεγιστοποιηθεί το $|S_{i_1} \cup \cdots \cup S_{i_k}|$).

1. Να δείξετε ότι ο άπληστος αλγόριθμος που σε κάθε επανάληψη επιλέγει το σύνολο που περιέχει τα περισσότερα ακάλυπτα στοιχεία έχει λόγο προσέγγισης $1 - (1 - \frac{1}{k})^k > 1 - \frac{1}{e}$.
2. Να δείξετε ότι για κάθε $\alpha \in (0, 1)$, η ύπαρξη ενός $(1 - \alpha)$-προσεγγιστικού αλγόριθμου για το Maximum Coverage οδηγεί σε έναν $\lceil \log_{1/\alpha} n \rceil$-προσεγγιστικό αλγόριθμο για το πρόβλημα Set Cover.

Άσκηση 7. Δίνεται ένα πλήρες μη κατευθυνόμενο γράφημα $G(V, E, \ell)$, κάθε ακμή e του οποίου έχει μήκος $\ell(e) \geq 0$. Υποθέτουμε ότι τα μήκη των ακμών ικανοποιούν την τριγωνική ανισότητα. Το ζητούμενο είναι ένας κύκλος Hamilton που το μέγιστο βάρος των ακμών του είναι το ελάχιστο δυνατόν. Να δείξετε ότι δεν υπάρχει προσεγγιστικός αλγόριθμος (πολυωνυμικού χρόνου) με λόγο προσέγγισης μικρότερο του 2 για αυτό το πρόβλημα (εκτός αν $\mathbf{P} = \mathbf{NP}$). Τι συμβαίνει αν τα μήκη των ακμών είναι αυθαίρετα;