



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Αλγόριθμοι και Πολυπλοκότητα

Διδάσκοντες: Άρης Παγουρτζής, Δώρα Σούλιου, Παναγιώτης Γροντάς

1η Σειρά Γραπτών Ασκήσεων - Ημ/νία Παράδοσης 11/11/2026

Άσκηση 1: Ασυμπτωτική πολυπλοκότητα

Σωστό ή λάθος; (αποδείξτε ή ανταποδείξτε)

(α) $f(n) + g(n) \in \Theta(\max(f(n), g(n)))$

(β) $f(n) \in O(g(n)) \Rightarrow 2^{f(n)} \in O(2^{g(n)})$

(γ) $f(n) \in \Theta(g(n)) \Rightarrow \log(f(n)) \in \Theta(\log(g(n)))$

(δ) $f(n) + o(f(n)) \in \Theta(f(n))$

Άσκηση 2: Αναγωγές μεταξύ προβλημάτων

Θεωρήστε τα εξής προβλήματα: (α) το πρόβλημα απόφασης Decision Perfect Matching (δίνεται γράφος G , υπάρχει perfect matching στον G); (β) το πρόβλημα εύρεσης Search Perfect Matching (δίνεται γράφος G , επιστρέφει ένα perfect matching στον G , αν υπάρχει, αλλιώς επιστρέφει $\perp$) και (γ) το πρόβλημα βελτιστοποίησης Maximum Matching (δίνεται γράφος G , επιστρέφει ένα matching του G μέγιστης πληθικότητας).

Διευκρινίστε τη σχέση μεταξύ αυτών των προβλημάτων. Συγκεκριμένα εξετάστε για το καθένα από αυτά τα προβλήματα το εξής ερώτημα: αν έχουμε αλγόριθμο πολυωνυμικού χρόνου για το πρόβλημα αυτό, ποια από τα άλλα δύο μπορούμε να λύσουμε σε πολυωνυμικό χρόνο και πώς;

Μπορείτε να ορίσετε κάποιο διαφορετικό πρόβλημα απόφασης τυχόν λύση του οποίου θα μας βοηθούσε να λύσουμε το Maximum Matching;

Άσκηση 3: Συναρτήσεις κατακερματισμού

Εξετάστε την μέθοδο κατακερματισμού ανοιχτής διευθυνσιοδότησης (open addressing) και υποθέστε ότι σε πίνακα m θέσεων έχουν εισαχθεί n κλειδιά με uniform hashing.

Αποδείξτε ότι στη συνέχεια ο μέσος χρόνος επιτυχούς αναζήτησης φράσσεται άνω από την ποσότητα $\frac{1}{\alpha} \ln \frac{1}{1-\alpha}$, όπου $\alpha = n/m$ ο παράγοντας φόρτου.

Υπόδειξη: ξεκινήστε με μια εκτίμηση για το αναμενόμενο πλήθος δοκιμών κατά την εισαγωγή του i -οστού κλειδιού.

Άσκηση 4 - Bonus: Συναρτήσεις κατακερματισμού

Υπενθύμιση: Μια οικογένεια συναρτήσεων κατακερματισμού $\mathcal{H} = \{h : U \rightarrow [m]\}$, όπου U ένα σύνολο κλειδιών, λέγεται *καθολική* αν

$$\forall x, y \in U, x \neq y : \Pr_{h \in \mathcal{H}}[h(x) = h(y)] \leq \frac{1}{m}$$

(σημείωση: χρησιμοποιούμε τον συμβολισμό $\mathbb{Z}_m = \{0, \dots, m-1\}$)

(α) Αποδείξτε ότι για $a \in [m] \setminus \{0\}$, $b \in [m]$ η οικογένεια συναρτήσεων $h_{a,b}(x) = (ax + b) \bmod m$ δεν έχει την ιδιότητα της καθολικότητας για $U = [t]$, $t > m$.

(β) Έστω ότι $U = [t], t > m$, και δίνεται πρώτος αριθμός $p > t$. Αποδείξτε ότι η οικογένεια συναρτήσεων $h_{a,b}(x) = ((ax + b) \bmod p) \bmod m, a \in [p] \setminus \{0\}, b \in [p]$, έχει την ιδιότητα της καθολικότητας.

(γ) Εξακολουθεί να ισχύει η ιδιότητα της καθολικότητας αν στο ερώτημα (β) θέσουμε $U = [p^2]$; Εξηγήστε την απάντησή σας.

Άσκηση 5: Union - Find

Σε ένα τμήμα υπάρχουν n φοιτητές τους οποίους αριθμούμε $1, \dots, n$. Κάθε φοιτητής i έχει μια μη-ταξινομημένη λίστα C_i με κωδικούς μαθημάτων επιλογής (αυθαίρετοι ακέραιοι), που μπορεί να είναι κενή. Έστω $L = |C_1| + \dots + |C_n|$ το συνολικό μήκος των λιστών. Δύο φοιτητές i, j λέγονται *συγγενικοί* αν υπάρχει ακολουθία $i = i_0, i_1, \dots, i_t = j$ ($t \geq 0$) στην οποία κάθε δύο διαδοχικοί φοιτητές έχουν τουλάχιστον ένα κοινό μάθημα. Η συγγένεια είναι σχέση ισοδυναμίας, και οι κλάσεις της λέγονται *ομάδες μελέτης*.

Παράδειγμα: έστω $C_1 = \{20, 30\}, C_2 = \{50, 70\}, C_3 = \{20, 50\}, C_4 = \{70\}, C_5 = \{30\}, C_6 = \{110\}$. Οι ομάδες μελέτης είναι $\{1, 2, 3, 4, 5\}$ και $\{6\}$. Προσέξτε ότι οι φοιτητές 1 και 4 δεν έχουν κοινό μάθημα, αλλά συνδέονται μέσω της ακολουθίας $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$.

- (α) Να διατυπώσετε αλγόριθμο που χρησιμοποιεί δομή Union-Find με στοιχεία τους φοιτητές και υπολογίζει το πλήθος των ομάδων και το μέγεθος της μεγαλύτερης σε χρόνο $O(L \log L + (L + n) \cdot \alpha(n))$. Να αποδείξετε ότι, όταν τελειώσει ο αλγόριθμος, για κάθε δύο φοιτητές i, j ισχύει $Find(i) = Find(j)$ αν και μόνο αν οι i, j είναι συγγενικοί.
- (β) Η γραμματοσειρά θέλει τα μεγέθη των k μεγαλύτερων ομάδων σε φθίνουσα σειρά. Να δείξετε πώς, μετά το (α), αυτό γίνεται σε χρόνο $O(n + k \log n)$. Πότε είναι αυτό καλύτερο από την ταξινόμηση όλων των μεγεθών;

Άσκηση 6: Μέτρηση αντιστροφών

Διατυπώστε αποδοτικό αλγόριθμο για το πρόβλημα της Μέτρησης Αντιστροφών: δίνονται n ακέραιοι $a_1, \dots, a_n$ και θέλουμε να μετρήσουμε πόσα ζεύγη είναι σε αντίστροφη σειρά, δηλαδή ισχύει $a_i > a_j$ και $i < j$.

Άσκηση 7: Υπολογισμός Κυρίαρχων Θέσεων (1.1 μον.)

Θεωρούμε πίνακα $A[1 \dots n]$ με n φυσικούς αριθμούς. Για κάθε $i = 2, \dots, n$, η θέση που *κυριαρχεί* της θέσης i στον πίνακα A είναι η πλησιέστερη θέση που προηγείται της i και η τιμή της ξεπερνά την τιμή $A[i]$. Τυπικά, θεωρώντας ότι $A[0] = \infty$, η θέση $B[i]$ που *κυριαρχεί* της θέσης i στον A είναι η μέγιστη θέση j , $0 \leq j < i$, για την οποία ισχύει ότι $A[j] > A[i]$ (βλ. ότι $B[1] = 0$). Να διατυπώσετε αποδοτικό αλγόριθμο που υπολογίζει τη θέση $B[i]$ που κυριαρχεί της θέσης i , για κάθε $i = 1, \dots, n$, στον πίνακα $A[1 \dots n]$. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 8: Ταξίδι με Ηλεκτρικό Αυτοκίνητο (2.2 μον.)

(α) Πρόκειται να πάμε ένα μεγάλο ταξίδι με το νέο ηλεκτρικό μας αυτοκίνητο 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 αποφύγουμε την ανάγκη στάσεων φόρτισης κατά τη διάρκεια της ημέρας, θέλουμε να προγραμματίσουμε το ταξίδι μας ώστε να ελαχιστοποιήσουμε τη μέγιστη απόσταση που θα διανύσουμε μέσα σε μία ημέρα (μπορούμε να σταματήσουμε για διανυκτέρευση και φόρτιση 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.

(β) Θεωρούμε δίκτυο n σταθμών φόρτισης ηλεκτρικών αυτοκινήτων κατά μήκος μιας (μεγάλης σε μήκος) οδικής αρτηρίας. Για ευκολία, θεωρούμε ότι η οδική αρτηρία είναι μονής κατεύθυνσης και τα αυτοκίνητα μετακινούνται μόνο από σταθμούς με μικρότερο δείκτη σε σταθμούς με μεγαλύτερο δείκτη. Τεχνολογικοί περιορισμοί επιτρέπουν στα αυτοκίνητα που φορτίζουν στον σταθμό i να φτάσουν το πολύ μέχρι τον σταθμό a_i , όπου $n \geq a_i \geq i+1$ (αυτό ανεξάρτητα από την ενέργεια που έχουν τα αυτοκίνητα όταν φτάνουν στον i – μάλιστα συμβαίνει συχνά να έχουμε $a_i > a_{i+1}$). Οι δείκτες $(a_1, \dots, a_{n-1})$ του μέγιστου σταθμού a_i όπου μπορούμε να φτάσουμε φορτίζοντας σε κάθε σταθμό i είναι γνωστοί (έχουμε ακόμη τετριμμένα ότι $a_n = n$).

Για τη διαχείριση του δικτύου χρειαζόμαστε μια δομή δεδομένων που με βάση τους δείκτες $(a_1, \dots, a_{n-1})$, υπολογίζει το ελάχιστο πλήθος φορτίσεων (και τους αντίστοιχους σταθμούς) για ένα αυτοκίνητο k με αφετηρία τον σταθμό s_k και προορισμό τον σταθμό t_k (όπου $1 \leq s_k < t_k \leq n$). Να περιγράψετε την αρχική επεξεργασία που απαιτείται στους δείκτες $(a_1, \dots, a_{n-1})$ και τον αλγόριθμο για την απάντηση των ερωτημάτων $\text{chrg}(s_k, t_k)$ σχετικά με το πλήθος και τους σταθμούς φόρτισης για να ταξιδέψουμε από τον σταθμό s_k στον σταθμό t_k . Να αιτιολογήσετε την ορθότητα του αλγορίθμου σας και την χρονική πολυπλοκότητα για την αρχική επεξεργασία και την απάντηση κάθε ερωτήματος $\text{chrg}(s_k, t_k)$.