Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Θεμελιώδη Θέματα Επιστήμης Υπολογιστών, 2023-24

2η σειρά γραπτών ασκήσεων

(λογική – υπολογισιμότητα – αλγοριθμικές τεχνικές
αριθμητικοί αλγόριθμοι – αλγόριθμοι γράφων)

Άσκηση 1. (Λογική και Αλγόριθμοι)

Διατυπώστε αποδοτικό αλγόριθμο που να δέχεται σαν είσοδο οποιονδήποτε τύπο της προτασιακής λογικής σε μορφή Horn και να αποφαίνεται αν είναι ικανοποιήσιμος. Σε περίπτωση που είναι θα πρέπει να επιστρέφει μία ανάθεση αληθοτιμών που ικανοποιεί τον τύπο.

Π.χ. με είσοδο $(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_4 \vee x_5 \vee \neg x_6) \wedge (x_6)$ θα πρέπει να επιστρέφει 'Ναι' και μία από τις αναθέσεις αληθοτιμών στις $(x_1, \ldots, x_6)$ που ικανοποιούν τον τύπο, π.χ. την ανάθεση $(\mathsf{True}, \mathsf{False}, \mathsf{True}, \mathsf{False}, \mathsf{True}, \mathsf{True})$ ενώ με είσοδο $(\neg x_1 \vee \neg x_2) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_2 \vee x_3) \wedge (x_2)$ θα πρέπει να επιστρέφει 'Όχι'.

Θεωρήστε ότι όλες οι μεταβλητές ενός τύπου δίνονται στη μορφή x_n, όπου n ένας φυσικός αριθμός.

Άσκηση 2. (Μηχανή Turing)

Να κατασκευαστεί μηχανή Turing με αλφάβητο $\Sigma = \{0, 1, \sqcup\}$ (δυαδική αναπαράσταση αριθμού) που να υπολογίζει τη συνάρτηση $x \to x - 1$.

Άσκηση 3. (Ελάχιστο πλήθος στάσεων)

Δίνεται κατευθυνόμενος γράφος με n κορυφές (οι πόλεις μιας χώρας) και m ακμές (δρόμους) που συνδέουν πόλεις μεταξύ τους. Δεν συνδέονται υποχρεωτικά όλες οι πόλεις με δρόμο. Μπορεί ανάμεσα σε δύο πόλεις να υπάρχουν δρόμοι και προς τις δύο κατευθύνσεις, αλλά δεν είναι απαραίτητο. Τα βάρη των ακμών αντιπροσωπεύουν τις αποστάσεις μεταξύ δύο πόλεων (αποστάσεις θετικές). Ένα αυτοκίνητο ξεκινά από την πόλη s και θέλει να φτάσει στην πόλη t. Το αυτοκίνητο έχει αυτονομία κίνησης, ως προς τη βενζίνη που μπορεί να αποθηκεύσει, k χιλιόμετρα. Σε κάθε πόλη υπάρχει η δυνατότητα ανεφοδιασμού (υπάρχει ακριβώς ένα βενζινάδικο). Δεδομένου ότι η καθυστέρηση για ανεφοδιασμό είναι μεγάλη επιθυμούμε να βρούμε το ελάχιστο πλήθος στάσεων για ανεφοδιασμό που χρειάζεται να κάνει το αυτοκίνητο προκειμένου να φτάσει στον προορισμό του, και την αντίστοιχη διαδρομή.

(α) Δώστε όσο το δυνατόν πιο αποδοτικό αλγόριθμο για το πρόβλημα αυτό. Εξηγήστε την ορθότητα και βρείτε την πολυπλοκότητα του αλγορίθμου που προτείνετε.

(β) Να σημειωθεί ότι δεν θέλουμε να βρούμε τη διαδρομή με την ελάχιστη συνολική απόσταση, αλλά αυτή *με τις λιγότερες δυνατές στάσεις*. Δώστε ένα παράδειγμα στο οποίο οι δύο αυτές διαδρομές να διαφέρουν.

Άσκηση 4. (2ο-ΕΣΔ) Θεωρούμε μη κατευθυνόμενο συνεκτικό γράφημα $G(V, E, w)$ με θετικά βάρη στις ακμές, και υποθέτουμε ότι $|E| \geq |V|$ και ότι όλα τα βάρη των ακμών του G είναι διαφορετικά μεταξύ τους. Έστω T το σύνολο όλων των συνδετικών δέντρων του G και έστω $t \in T$ ένα ελάχιστο συνδετικό δέντρο του G. Το Δεύτερο Ελαφρύτερο Συνδετικό Δέντρο (2ο-ΕΣΔ) του G είναι ένα συνδετικό δέντρο $t' \in T$ τέτοιο ώστε $w(t') = min_{t'' \in T \setminus \{t\}}\{w(t'')\}$. Με άλλα λόγια, το δεύτερο ελάχιστο

συνδετικό δέντρο t' είναι ένα συνδετικό δέντρο του G που έχει βάρος μεγαλύτερο ή ίσο από το βάρος του ΕΣΔ t και μικρότερο ή ίσο από το βάρος κάθε άλλου συνδετικού δέντρου.

1. Να δείξετε ότι το ΕΣΔ του G είναι μοναδικό, και ότι κάτι τέτοιο δεν ισχύει απαραίτητα για το 2ο-ΕΣΔ του G.

2. Έστω t το ΕΣΔ του G και $t^{'}$ το 2ο-ΕΣΔ.Να δείξετε ότι τα t και $t^{'}$ διαφέρουν κατά μία μόνο ακμή, δηλαδή ότι υπάρχουν ακμές $e \in t$ και $e^{'} \notin t$ τέτοιες ώστε $t^{'} = t \cup \{e^{'}\} \setminus \{e\}$

3. Έστω t ένα συνδετικό δέντρο του G και, για οποιοδήποτε ζεύγος κορυφών $u, v \in V$, έστω e_{uv}^{max} η ακμή μεγίστου βάρους στο στο μοναδικό $u - v$ μονοπάτι στο t. Να διατυπώσετε αλγόριθμο χρόνου $O(|V|^2)$ που δέχεται ως είσοδο το t και προσδιορίζει την ακμή e_{uv}^{max} για όλα τα ζεύγη κορυφών $u, v \in V$.

4. Να διατυπώσετε αποδοτικό αλγόριθμο που να υπολογίζει το 2ο-ΕΣΔ του G. Εξηγήστε την ορθότητα και βρείτε την πολυπλοκότητα του αλγορίθμου που προτείνετε.

Άσκηση 5. (r-περιορισμένο μονοπάτι)
Έστω μη κατευθυνόμενο γράφημα $G(V, E, w)$ με θετικά βάρη w στις ακμές, και έστω $s, t \in V$. Για κάποιο $r > 0$, λέμε ότι ένα $s - t$ μονοπάτι p είναι r-περιορισμένο αν το βάρος κάθε ακμής στο p είναι μικρότερο ή ίσο του r.

1. Να διατυπώσετε αποδοτικό αλγόριθμο που για δεδομένο r, ελέγχει αν υπάρχει r-περιορισμένο s-t μονοπάτι στο G.

2. Να δείξετε ότι το G περιέχει r-περιορισμένο s-t μονοπάτι αν και μόνο αν ένα ελάχιστο Συνδετικό Δέντρο του G περιέχει r-περιορισμένο s-t μονοπάτι.

3. Να διατυπώσετε αποδοτικό αλγόριθμο που υπολογίζει την ελάχιστη τιμή του r για την οποία υπάρχει r-περιορισμένο μονοπάτι s-t στο G.

Άσκηση 6. (Αναδρομή – Επανάληψη – Επαγωγή)
(α) Εκφράστε το πλήθος κινήσεων δίσκων που κάνει ο αναδρομικός αλγόριθμος για τους πύργους του Hanoi, σαν συνάρτηση του αριθμού των δίσκων n.
(β) Δείξτε ότι οι κινήσεις δίσκων του αναδρομικού αλγορίθμου είναι ακριβώς ίδιες με τις κινήσεις δίσκων του επαναληπτικού αλγορίθμου (με κατάλληλη αρίθμηση των πασσάλων).
Προσοχή: θα πρέπει να ορίσετε προσεκτικά την 'θετική φορά' ώστε να ισχύει αυτό.
(γ) Δείξτε ότι ο αριθμός των κινήσεων των παραπάνω αλγορίθμων είναι ο ελάχιστος μεταξύ όλων των δυνατών αλγορίθμων για το πρόβλημα αυτό.
(δ) Θεωρήστε το πρόβλημα των πύργων του Hanoi με 4 αντί για 3 πασσάλους. Σχεδιάστε αλγόριθμο μετακίνησης n δίσκων από τον πάσσαλο 1 στον πάσσαλο 4 ώστε το πλήθος των κινήσεων να είναι σημαντικά μικρότερο από το πλήθος των κινήσεων που απαιτούνται όταν υπάρχουν μόνο 3 πάσσαλοι. Εκφράστε τον αριθμό των απαιτούμενων κινήσεων σαν συνάρτηση του n.

Άσκηση 7. (Εύρεση ΜΚΔ) Θεωρήστε τον παρακάτω αλγόριθμο για εύρεση ΜΚΔ που είναι γνωστός ως Binary GCD.

$\mathsf{bgcd}(a, b)$: (* υποθέτουμε $a, b > 0$ *)

- Αν $a = b$ επίστρεψε a
- αν a, b άρτιοι επίστρεψε $2 \cdot \mathsf{bgcd}(a/2, b/2)$
- αν a είναι άρτιος και b περιττός επίστρεψε $\mathsf{bgcd}(a/2, b)$, και αντίστοιχα αν b άρτιος και a περιττός
- αν a, b περιττοί επίστρεψε $\mathsf{bgcd}(\min(a, b), |a - b|/2)$

(α) Αποδείξτε την ορθότητα του Binary GCD.

(β) Ποια είναι η πολυπλοκότητά του και γιατί;

(γ) Υλοποιήστε τον και συγκρίνετε την απόδοτικότητά του με αυτήν του Ευκλείδειου αλγόριθμου. Δοκιμάστε τους δύο αλγορίθμους με τουλάχιστον 10 ζεύγη πολύ μεγάλων αριθμών.

Άσκηση 8. (Επαναλαμβανόμενος Τετραγωνισμός – Κρυπτογραφία)

(α) Γράψτε πρόγραμμα σε γλώσσα της επιλογής σας (θα πρέπει να υποστηρίζει πράξεις με αριθμούς 100δων ψηφίων) που να ελέγχει αν ένας αριθμός είναι πρώτος με τον έλεγχο (test) του Fermat:

Αν n πρώτος τότε για κάθε a τ.ώ. $1 < a < n - 1$, ισχύει

$$a^{n-1} \bmod n = 1$$

Αν λοιπόν, για δεδομένο n βρεθεί a ώστε να μην ισχύει η παραπάνω ισότητα τότε ο αριθμός n είναι οπωσδήποτε σύνθετος. Αν η ισότητα ισχύει για το συγκεκριμένο a, τότε η δοκιμή πρέπει να επαναληφθεί με νέο a, καθώς υπάρχει περίπτωση ο αριθμός να είναι σύνθετος και παρ'όλα αυτά η ισότητα να ισχύει για κάποιες τιμές του a. Μια ενδιαφέρουσα ιδιότητα λέει ότι, αν ο n είναι σύνθετος, η πιθανότητα να ισχύει η ισότητα είναι $\leq 1/2$ (αυτό ισχύει για όλα τα n εκτός από κάποιες 'παθολογικές' περιπτώσεις, που λέγονται αριθμοί Carmichael, δείτε Σημ. 2 παρακάτω). Έτσι, μπορούμε να αυξήσουμε σημαντικά την πιθανότητα επιτυχίας (δηλ. της επιβεβαίωσης της συνθετότητας του αριθμού n) επαναλαμβάνοντας μερικές φορές τη δοκιμή (τυπικά 30 φορές) με διαφορετικό a. Αν όλες τις φορές βρεθεί να ισχύει η παραπάνω ισότητα τότε λέμε ότι το n "περνάει το test" και ανακηρύσσουμε το n πρώτο αριθμό· αν έστω και μία φορά αποτύχει ο έλεγχος, τότε είμαστε βέβαιοι ότι ο αριθμός είναι σύνθετος.

Το πρόγραμμά σας θα πρέπει να δουλεύει σωστά για αριθμούς χιλιάδων ψηφίων. Δοκιμάστε την με τους αριθμούς:

$67280421310721, 1701411834604692317316873037158841057 21, 2^{2281} - 1, 2^{9941} - 1$

Σημείωση 1: το $a^{2^{9941}-2}$ έχει 'αστρονομικά' μεγάλο πλήθος ψηφίων (δεν χωράει να γραφτεί ούτε σε ολόκληρο το σύμπαν!), ενώ το $a^{2^{9941}-2} \bmod (2^{9941} - 1)$ είναι σχετικά "μικρό" (έχει μερικές χιλιάδες δεκαδικά ψηφία *μόνο* :-)) οπότε είναι δυνατόν να το υπολογίσουμε (με λίγη προσοχή).

Σημείωση 2: Υπάρχουν (λίγοι) σύνθετοι που έχουν την ιδιότητα να περνούν τον έλεγχο Fermat για κάθε a που είναι σχετικά πρώτο με το n, οπότε για αυτούς το test θα αποτύχει όσες δοκιμές και αν γίνουν (εκτός αν πετύχουμε κατά τύχη a που δεν είναι σχετικά πρώτο με το n, πράγμα αρκετά απίθανο για αρκετά μεγάλο n). Αυτοί οι αριθμοί λέγονται *Carmichael* – δείτε και `http://en.wikipedia.org/wiki/Carmichael_number`. Ελέγξτε τη συνάρτησή σας με *αρκετά μεγάλους* αριθμούς Carmichael που θα βρείτε π.χ. στη σελίδα `http://de.wikibooks.org/wiki/Pseudoprimzahlen:_Tabelle_Carmichael-Zahlen`. Τι παρατηρείτε;

(β) Μελετήστε και υλοποιήστε τον έλεγχο Miller-Rabin (π.χ. από τις σημειώσεις που θα βρείτε στη σελίδα του μαθήματος στο Helios) που αποτελεί βελτίωση του ελέγχου του Fermat και δίνει σωστή απάντηση με πιθανότητα τουλάχιστον 1/2 για *κάθε* φυσικό αριθμό (οπότε με 30 επαναλήψεις έχουμε αμελητέα πιθανότητα λάθους για κάθε αριθμό εισόδου). Δοκιμάστε τον με διάφορους αριθμούς Carmichael. Βλέπετε κάτι περίεργο; Πώς το εξηγείτε;

(γ) Γράψτε πρόγραμμα που να βρίσκει όλους τους πρώτους αριθμούς Mersenne, δηλαδή της μορφής $n = 2^x - 1$ με $1 < x < 200$ (σημειώστε ότι αν το x δεν είναι πρώτος, ούτε το $2^x - 1$ είναι πρώτος – μπορείτε να το αποδείξετε;). Αντιπαραβάλετε με όσα αναφέρονται στην ιστοσελίδα `https://www.mersenne.org/primes/`.

Προθεσμία υποβολής και οδηγίες. Οι απαντήσεις θα πρέπει να υποβληθούν έως τις 27/12/2023, και ώρα 23:59, σε ηλεκτρονική μορφή, στο Helios (προσπαθήστε το τελικό αρχείο να είναι μεγέθους <5MB συνολικά). Αποδεκτά format: pdf, png, jpg, gif, και zip ή gz που να περιέχει κάποια από τα προηγούμενα.

Δεν θα διορθωθούν εργασίες που στέλνονται με email.

Τα ερωτήματα με (*) είναι προαιρετικά. Εφ'όσον τα επιλύσετε μπορούν να προσμετρηθούν στη θέση ερωτημάτων που δεν απαντήσατε.

Συνιστάται *θερμά* να αφιερώσετε ικανό χρόνο για να λύσετε τις ασκήσεις μόνοι σας προτού καταφύγετε σε οποιαδήποτε *θεμιτή* βοήθεια (διαδίκτυο, βιβλιογραφία, συζήτηση με συμφοιτητές). Σε κάθε περίπτωση, οι απαντήσεις θα πρέπει να είναι *αυστηρά* ατομικές και να περιλαμβάνουν αναφορές σε κάθε πηγή που χρησιμοποιήσατε.

Για να βαθμολογηθείτε θα πρέπει να παρουσιάσετε σύντομα τις λύσεις σας σε ημέρα και ώρα που θα ανακοινωθεί αργότερα. Για απορίες / διευκρινίσεις: στείλτε μήνυμα στη διεύθυνση `focs@corelab.ntua.gr`.