Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Θεμελιώδη Θέματα Επιστήμης Υπολογιστών, 2022-23

2η σειρά γραπτών ασκήσεων

(αλγοριθμικές τεχνικές – αριθμητικοί αλγόριθμοι
αλγόριθμοι γράφων)

Άσκηση 1. (Αναδρομή – Επανάληψη – Επαγωγή)

(α) Εκφράστε τον αριθμό κινήσεων δίσκων που κάνει ο αναδρομικός αλγόριθμος για τους πύργους του Hanoi, σαν συνάρτηση του αριθμού των δίσκων n.

(β) Δείξτε ότι ο αριθμός κινήσεων του αναδρομικού ισούται με τον αριθμό μετακινήσεων του επαναληπτικού αλγορίθμου.

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 βημάτων να είναι σημαντικά μικρότερο από το πλήθος των βημάτων που απαιτούνται όταν υπάρχουν μόνο 3 πάσσαλοι. Εκφράστε τον αριθμό των απαιτούμενων βημάτων σαν συνάρτηση του n.

Άσκηση 2. (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:

$67280421310721, 170141183460469231731687303715884105721, 2^{2281}-1, 2^{9941}-1$

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/`.

Άσκηση 3. (Αριθμοί Fibonacci)

(α) Υλοποιήστε και συγκρίνετε τους εξής αλγορίθμους για υπολογισμό του n-οστού αριθμού Fibonacci: αναδρομικό με memoization, επαναληπτικό, και με πίνακα.

Υλοποιήστε τους αλγορίθμους σε γλώσσα που να υποστηρίζει πολύ μεγάλους ακεραίους (100δων ψηφίων), π.χ. σε Python. Χρησιμοποιήστε τον πολλαπλασιασμό ακεραίων που παρέχει η γλώσσα. Τι συμπεραίνετε;

(β) Δοκιμάστε να λύσετε το παραπάνω πρόβλημα με ύψωση σε δύναμη, χρησιμοποιώντας τη σχέση του F_n με το ϕ (χρυσή τομή). Τι παρατηρείτε;

(γ) Υλοποιήστε συνάρτηση που να δέχεται σαν είσοδο δύο θετικούς ακεραίους n, k και να υπολογίζει τα k λιγότερο σημαντικά ψηφία του n-οστού αριθμού Fibonacci.

(δ*) Αναζητήστε και εξετάστε τη μέθοδο Fast Doubling σε σχέση με τα παραπάνω ερωτήματα. Συγκρίνετέ την με τη μέθοδο του πίνακα θεωρητικά και υπολογιστικά.

Άσκηση 4. (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 5. (Σχεδόν Δέντρο)

Έστω συνεκτικό μη κατευθυνόμενο γράφημα $G(V, E, w)$ με θετικά βάρη w στις ακμές. Υποθέτουμε ότι το G είναι σχεδόν δέντρο, με την έννοια ότι $|E| = |V| + c$, για κάποια σταθερά c. Να διατυπώσετε αποδοτικό αλγόριθμο (κατά προτίμηση γραμμικού χρόνου) για τον υπολογισμό ενός ελάχιστου συνδετικού δέντρου σε ένα τέτοιο γράφημα G. Να αιτιολογήσετε την ορθότητα και την υπολογιστική πολυπλοκότητα του αλγορίθμου σας.

Άσκηση 6. (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 7. (Μονοπάτι με ελάχιστο πλήθος ακμών)

Θεωρούμε κατευθυνόμενο γράφημα $G(V, E, w)$ με n κορυφές, m ακμές και θετικά μήκη w στις ακμές, και μια αρχική κορυφή s του G. Να διατυπώσετε αποδοτικό αλγόριθμο που για όλες τις κορυφές $u \in V$, υπολογίζει ένα συντομότερο $s - u$ μονοπάτι με ελάχιστο πλήθος ακμών (μεταξύ όλων των συντομότερων $s - u$ μονοπατιών). Ποια η υπολογιστική πολυπλοκότητα του αλγορίθμου σας;

Άσκηση 8. (Επιβεβαίωση Αποστάσεων)

Θεωρούμε ένα κατευθυνόμενο γράφημα $G(V, E, \ell)$ με n κορυφές, m ακμές και (ενδεχομένως αρνητικό) μήκος $\ell(e)$ σε κάθε ακμή του $e \in E$. Συμβολίζουμε με $d(u, v)$ την απόσταση των κορυφών u και v (δηλ. το $d(u, v)$ είναι ίσο με το μήκος της συντομότερης διαδρομής από την u στην v) στο G. Δίνονται n αριθμοί $\delta_1, \ldots, \delta_n$, όπου κάθε δ_k (υποτίθεται ότι) ισούται με την απόσταση $d(v_1, v_k)$ στο G. Να διατυπώσετε αλγόριθμο που σε χρόνο $\Theta(n + m)$, δηλ. γραμμικό στο μέγεθος του γραφήματος, ελέγχει αν τα $\delta_1, \ldots, \delta_n$ πράγματι ανταποκρίνονται στις αποστάσεις των κορυφών από την v_1, δηλαδή αν για κάθε $v_k \in V$, ισχύει ότι $\delta_k = d(v_1, v_k)$. Αν αυτό αληθεύει, ο αλγόριθμός σας πρέπει να υπολογίζει και να επιστρέφει ένα Δέντρο Συντομότερων Μονοπατιών με ρίζα τη v_1 (χωρίς ο χρόνος εκτέλεσης να ξεπεράσει το $\Theta(n + m)$).

Προθεσμία υποβολής και οδηγίες. Οι απαντήσεις θα πρέπει να υποβληθούν έως τις 27/12/2022, 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.

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`.