Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Θεμελιώδη Θέματα Επιστήμης Ηλεκτρονικών Υπολογιστών, 2022-23

Επιμέλεια ασκήσεων: Ε. Ζάχος, Α. Παγουρτζής, Π. Ποτίκας, Α. Χαλκή

3η σειρά γραπτών ασκήσεων

(αριθμητικοί υπολογισμοί - αλγοριθμικές τεχνικές – αλγόριθμοι γράφων και δικτύων – εφαρμογές)

Άσκηση 1. (Αριθμητικοί υπολογισμοί)

Υλοποιήστε σε γλώσσα της επιλογής σας τους 3 αλγορίθμους για υπολογισμό του n-οστού αριθμού Fibonacci που κάναμε στο μάθημα. Εκτελέστε τα προγράμματά σας για διάφορες τιμές του n (μέχρι και 10^6) και μετρήστε τον χρόνο υπολογισμού που χρειάζεται ο κάθε αλγόριθμος. Τι παρατηρείτε; Εξηγήστε τα αποτελέσματα.

Άσκηση 2. (Αναγωγές προς επίλυση προβλημάτων / αλγόριθμοι γράφων)

Ορίζουμε το πρόβλημα της Διαδρομής Ίππου ως εξής: είσοδος είναι οι διαστάσεις ορθογώνιας σκακιέρας n, m, καθώς και κάποια απαγορευμένα τετράγωνα που δίνονται σαν ζεύγη (i, j). Σκοπός είναι να βρούμε αν ένα άλογο (ίππος) που ξεκινά από το αρχικό τετράγωνο $(1, 1)$ μπορεί να φτάσει στο τελικό τετράγωνο (n, m) χωρίς να χρησιμοποιήσει απαγορευμένα τετράγωνα, και αν ναι με ποιον τρόπο. (Κίνηση αλόγου: 2 τετράγωνα προς μία κατεύθυνση, οριζόντια ή κάθετα, και μετά 1 τετράγωνο αριστερά ή δεξιά, κάθετα στην αρχική κατεύθυνση).

(α) Έχει λύση το πρόβλημα για τη σκακιέρα 3×5 με απαγορευμένα τετράγωνα τα $(2, 2)$ και $(3, 4)$; Αν ναι, ποιά;

(β) Περιγράψτε με σαφήνεια έναν όσο το δυνατόν πιο αποδοτικό αλγόριθμο που να επιλύει το πρόβλημα της Διαδρομής Ίππου στη γενική περίπτωση. Ποια είναι η πολυπλοκότητα του αλγορίθμου σας σε σχέση με τα n, m και το πλήθος w των απαγορευμένων τετραγώνων;

Υπόδειξη: προσπαθήστε να μετατρέψετε το πρόβλημα σε κάποιο γνωστό σας πρόβλημα για το οποίο υπάρχει αποδοτικός αλγόριθμος. Μπορείτε να χρησιμοποιήσετε γνωστούς αλγορίθμους σαν υπορουτίνες, εάν όμως χρειάζονται τροποποιήσεις θα πρέπει να τις αναφέρετε συνοπτικά.

(γ) Πώς πρέπει να τροποποιήσετε / συμπληρώσετε τον αλγόριθμό σας ώστε να επιστρέφει τη διαδρομή με το ελάχιστο πλήθος κινήσεων του ίππου (αν βέβαια υπάρχει διαδρομή); Αλλάζει η πολυπλοκότητα του αλγορίθμου σας; Εξηγήστε.

(δ) Τροποποιήστε (συμπληρώνοντας ή επανασχεδιάζοντας) τη διαδικασία `bfs` ώστε να δέχεται επιπλέον μία παράμετρο k και να δίνει στην έξοδο το πλήθος των κόμβων που έχουν απόσταση από τον αρχικό κόμβο μικρότερη ή ίση του k.

Άσκηση 3. (Αντιπροσωπεία Κινητών Τηλεφώνων)

Επιθυμούμε να βελτιστοποιήσουμε την λειτουργία μιας νέας αντιπροσωπείας κινητών για τις επόμενες n ημέρες. Για κάθε ημέρα $i, 1 \leq i \leq n$, υπάρχει μια (απόλυτα ασφαλής) πρόβλεψη d_i του αριθμού των πωλήσεων. Όλες οι πωλήσεις λαμβάνουν χώρα το μεσημέρι, και όσα κινητά δεν πωληθούν, αποθηκεύονται. Υπάρχει δυνατότητα αποθήκευσης μέχρι S κινητών και το κόστος είναι C για κάθε κινητό που αποθηκεύεται και για κάθε ημέρα αποθήκευσης. Το κόστος μεταφοράς για την προμήθεια νέων κινητών είναι K ευρώ, ανεξάρτητα από το πλήθος των κινητών που προμηθευόμαστε και τα νέα κινητά φθάνουν λίγο πριν το μεσημέρι (άρα αν πωληθούν αυθημερόν, δεν χρειάζονται αποθήκευση). Αρχικά δεν υπάρχουν καθόλου κινητά στην αντιπροσωπεία. Το ζητούμενο είναι να προσδιορισθούν οι παραγγελίες (δηλ. πόσα κινητά θα παραγγείλουμε και πότε) ώστε να ικανοποιηθούν οι προβλεπόμενες πωλήσεις με το ελάχιστο δυνατό συνολικό κόστος (αποθήκευσης και μεταφοράς). Να διατυπώσετε αλγόριθμο δυναμικού προγραμματισμού με χρόνο εκτέλεσης πολυωνυμικό στο nS για τη βελτιστοποίηση της λειτουργίας της αντιπροσωπείας.

Άσκηση 4. *(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, δεδομένου ενός n βρεθεί a ώστε να μην ισχύει η παραπάνω ισότητα τότε ο αριθμός n είναι οπωσδήποτε σύνθετος. Αν η ισότητα ισχύει, τότε το n είναι πρώτος με μεγάλη πιθανότητα (για τους περισσότερους αριθμούς $\geq 1/2$). Για να αυξήσουμε σημαντικά την πιθανότητα μπορούμε να επαναλάβουμε μερικές φορές (τυπικά 40 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 ο αριθμός είναι σύνθετος.

Η συνάρτησή σας θα πρέπει να δουλεύει σωστά για αριθμούς έως και 1000 ψηφίων, π.χ. να μπορεί να ελέγξει αν ο αριθμός $2^x - 1$ είναι πρώτος για κάθε $x \leq 1000$. Δοκιμάστε την για $x = 100i, 1 \leq i \leq 10$.

(β) Δοκιμάστε τη συνάρτησή σας με διάφορους γνωστούς πρώτους αριθμούς, π.χ. δείτε:

http://en.wikipedia.org/wiki/List_of_prime_numbers, καθώς και με σύνθετους αριθμούς (μπορείτε να κατασκευάσετε μεγάλους σύνθετους ως γινόμενα 2 πρώτων).

Σημείωση: θυμηθείτε ότι το $a^{2^{1000}-2}$ έχει αστρονομικά μεγάλο πλήθος ψηφίων (δεν χωράει να γραφτεί σε ολόκληρο το σύμπαν!), ενώ το $a^{2^{1000}-2} \bmod (2^{1000}-1)$ είναι σχετικά "μικρό" (έχει μερικές εκατοντάδες δεκαδικά ψηφία *μόνο* :-)

Προθεσμία υποβολής και οδηγίες. Οι απαντήσεις θα πρέπει να υποβληθούν έως τις 28/12/2022, στις 23:59, σε ηλεκτρονική μορφή, στο helios (φροντίστε το τελικό αρχείο να είναι μεγέθους < 10MB συνολικά).

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 (δηλαδή όχι 'copy-paste').

Τα ερωτήματα με αστεράκι (*) είναι προαιρετικά και μετράνε σαν μπόνους.

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 `ppotik@cs.ntua.gr`.