ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ - ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
Ειδικά Θέματα Αλγορίθμων
Διδάσκων: Ε. Μαρκάκης, Φθινοπωρινό εξάμηνο 2024

1η σειρά ασκήσεων **Προθεσμία παράδοσης: 29 Νοεμβρίου 2024**

Πρόβλημα 1. (10 μονάδες) Έστω ότι θέλετε να λύσετε ένα πρόβλημα πάνω σε έναν πίνακα n στοιχείων, και θέλετε να επιλέξετε τον καλύτερο από τους παρακάτω 2 αλγορίθμους. Μπορείτε να υποθέσετε ότι το n είναι δύναμη του 2.

- Ο αλγόριθμος Α δημιουργεί 3 υποπροβλήματα σε υποπίνακες με το μισό μέγεθος από το αρχικό ο καθένας, μετά τα λύνει αναδρομικά, και στο τέλος συνδυάζει τις επιμέρους λύσεις για να δημιουργήσει μια λύση του αρχικού προβλήματος σε γραμμικό χρόνο.
- Ο αλγόριθμος B δημιουργεί 2 υποπροβλήματα με μέγεθος $n-1$, και στο τέλος συνδυάζει τις επιμέρους λύσεις για να δημιουργήσει μια λύση στο αρχικό πρόβλημα σε σταθερό χρόνο.

Επίσης, όταν $n = 1$, και οι 2 αλγόριθμοι λύνουν το πρόβλημα σε σταθερό χρόνο. Ποιον από τους 2 αλγορίθμους θα επιλέξετε;

Πρόβλημα 2. (26 μονάδες)

(i) (10 μονάδες) Η εξίσωση $7 \cdot x \equiv 1 \bmod 128$ έχει λύση εντός του Z_{128}; Αν ναι, βρείτε την. Να επαναλάβετε την ερώτηση με το 126 αντί του 128.

(ii) (8 μονάδες) Χωρίς να χρησιμοποιήσετε κομπιουτεράκι ή οποιοδήποτε άλλο μέσο (ούτε τον αλγόριθμο επαναλαμβανόμενου τετραγωνισμού), υπολογίστε τις ποσότητες: $2 \cdot 3^{68}\ mod\ 23$, και $(2 \cdot 6^{48} + 11^{49})\ mod\ 35$.

(iii) (8 μονάδες) Θεωρήστε το Fermat test που παρουσιάστηκε στις διαφάνειες σαν ένας αλγόριθμος για το primality testing. Έστω ότι ένας αριθμός n είναι σύνθετος και δεν είναι αριθμός Carmichael. Έστω επίσης ότι για να τρέξετε το Fermat test, επιλέγετε τυχαία έναν αριθμό α, έτσι ώστε $\alpha \in \{1, \ldots, n-1\}$ και $gcd(\alpha, n) = 1$. Να δείξετε ότι τουλάχιστον για τις μισές από τις πιθανές επιλογές για το α, ο αριθμός n δεν θα περάσει το Fermat test (δηλαδή θα αναγνωρισθεί σωστά ως σύνθετος).

Πρόβλημα 3. (16 μονάδες) Έστω ότι σε ένα σύστημα RSA, ο Oscar υποκλέπτει το ciphertext, το οποίο είναι ίσο με $C = 10$. Αν ξέρει ότι το ciphertext αυτό είχε σταλεί σε ένα χρήστη με public key $e = 13$, και $n = 35$, ποιο ήταν το plaintext? Δείξτε αναλυτικά όλα τα βήματα που πρέπει να ακολουθήσετε για να βρείτε το μήνυμα. Αν χρειαστεί να υψώσετε σε δύναμη για τους υπολογισμούς που θα κάνετε, θα πρέπει να χρησιμοποιήσετε τον αλγόριθμο επαναλαμβανόμενου τετραγωνισμού.

Πρόβλημα 4. (16 μονάδες) Έστω ένας πίνακας Α n θέσεων, από το 1 ως το n. Θεωρήστε τον παρακάτω ψευδοκώδικα, που παίρνει ως είσοδο έναν ακέραιο m με $1 \leq m \leq n$.

```
int foo(int m)
if (m ≤ √n) return n + 10
else {
    int x=0
    for i = 1 to n    {x = x + A[i]}
    return x
    }
```

(i) (8 μονάδες) Αναλύστε την πολυπλοκότητα καλύτερης, χειρότερης και μέσης περίπτωσης. Για την ανάλυση της μέσης περίπτωσης, θεωρήστε ότι ο ακέραιος m επιλέγεται τυχαία με ομοιόμορφη κατανομή από τους ακεραίους $1, 2, \ldots, n$.

(ii) (8 μονάδες) Απαντήστε στα ίδια ερωτήματα αν τώρα αντικαταστήσουμε την συνθήκη του if με if $(m \leq n/2)$.

Πρόβλημα 5. (14 μονάδες) Τα παρακάτω δύο υποερωτήματα σχετίζονται με το πρόβλημα μεγιστοποίησης ροής.

(i) (8 μονάδες) Θεωρήστε την ακόλουθη παραλλαγή του προβλήματος μέγιστης ροής. Μας δίνεται και πάλι ένας γράφος με έναν κόμβο-αφετηρία και έναν κόμβο προορισμό, αλλά εκτός από τις χωρητικότητες των ακμών, υπάρχει επιπλέον και μια χωρητικότητα $c(v)$, για κάθε κόμβο $v \in V$. Επομένως, για να είναι μια ροή εφικτή, εκτός από τους περιορισμούς που πρέπει να ισχύουν, όπως και στο αρχικό πρόβλημα, θα πρέπει επίσης να ικανοποιείται ότι η ροή που διέρχεται από έναν κόμβο v δεν μπορεί να υπερβαίνει το $c(v)$. Δείξτε ότι αυτή η έκδοση μπορεί να αναχθεί στην επίλυση του αρχικού προβλήματος. Ουσιαστικά θα πρέπει να μετατρέψετε τον αρχικό γράφο G, που έχει χωρητικότητες και στις ακμές και στους κόμβους, σε ένα νέο γράφο G' που έχει χωρητικότητες μόνο σε ακμές, έτσι ώστε η μέγιστη ροή στον G' να ισούται με τη μέγιστη ροή στον G.

(ii) (6 μονάδες) Θεωρήστε έναν γράφο $G = (V, E)$, χωρίς βάρη, και έστω s, t 2 κορυφές του γράφου. Θέλουμε να βρούμε τον ελάχιστο αριθμό από ακμές που αν τις αφαιρέσουμε ο υπολειπόμενος γράφος δεν έχει μονοπάτι από την s στην t. Δείξτε πώς μπορείτε να λύσετε αυτό το πρόβλημα χρησιμοποιώντας τους αλγορίθμους που έχουμε δει για μεγιστοποίηση της ροής.

Πρόβλημα 6. (18 μονάδες)

(i) (8 μονάδες) Θεωρήστε την ακολουθία των αριθμών Fibonacci, με $F_n = F_{n-1} + F_{n-2}$, $F_1 = 1$, $F_0 = 0$. Αποδείξτε ότι ισχύει η παρακάτω σχέση για κάθε $k \geq 1$.

$$F_{n+k} = F_{n+1} \cdot F_k + F_n \cdot F_{k-1}$$

(ii) (10 μονάδες) Χρησιμοποιήστε την παραπάνω σχέση για να σχεδιάσετε έναν αλγόριθμο για τον υπολογισμό του n-οστού αριθμού Fibonacci που απαιτεί $O(logn)$ πλήθος από αριθμητικές πράξεις (προσθέσεις και πολλαπλασιασμούς αριθμών). Επιχειρηματολογήστε αν ο αλγόριθμος αυτός είναι πιο γρήγορος στην πράξη από τον άλλο αλγόριθμο που έχουμε δει με πολυπλοκότητα $O(logn)$.