



ΕΞΕΤΑΣΗ ΜΑΘΗΜΑΤΟΣ
Ανάλυση Αλγορίθμων
24 Ιουνίου 2022

Ονοματεπώνυμο:

A.M.:

Χρόνος Εξέτασης: 2 ώρες

ΘΕΜΑ 1. (25 μον.)

A. (5 μον.)

Έστω δύο αλγόριθμοι A και B με χρόνο εκτέλεσης $T_A(n) = 4n \log n$ και $T_B(n) = 20n$. Ποιος αλγόριθμος είναι πιο αποδοτικός ως προς την χρονική πολυπλοκότητα \mathcal{O} ; Βρείτε το n_0 όπου για κάθε $n > n_0$ ο αλγόριθμος που επιλέξατε είναι πιο αποδοτικός από τον άλλον.

B. (10 μον.)

Έστω η συνάρτηση:

$$f(x) = \sum_{i=0}^n a_i x^i$$

Αποδείξτε ότι $f(x) = \mathcal{O}(x^n)$.

Γ. (10 μον.)

Δίνεται ο παρακάτω κώδικας σε γλώσσα προγραμματισμού C:

```
int k=0;
while (k < n/2) {
    for (int l = k+1; l <= n; l=l*2)
        calc(k,l);
    k = k+2;
}
```

Υπολογίστε πόσες φορές θα κληθεί η συνάρτηση calc και ποια είναι η υπολογιστική πολυπλοκότητα του κώδικα εάν θεωρήσουμε ότι η calc έχει σταθερό χρόνο εκτέλεσης ($\mathcal{O}(1)$);

ΘΕΜΑ 2. (25 μον.)

A. (15 μον.)

Ποια/ες από τις παρακάτω σχέσεις δεν μπορεί/ούν να επιλυθεί/ούν με την Κύρια Μέθοδο (Master Method);

i) $T(n) = 16T\left(\frac{n}{4}\right) + n$

ii) $T(n) = T\left(\frac{n}{5}\right) + 5$

iii) $T(n) = 2T\left(\frac{n}{2}\right) - n \log n$

Δικαιολογήστε την απάντησή σας σε κάθε περίπτωση.

B. (10 μον.)

Υλοποιήστε μια συνάρτηση (σε μορφή ψευδοκώδικα ή γλώσσας προγραμματισμού C) η οποία θα υπολογίζει τη δύναμη 5^n , $n \in \mathbb{N}$, με υπολογιστική πολυπλοκότητα $\mathcal{O}(\lg n)$.

ΘΕΜΑ 3. (25 μον.)**A. (10 μον.)**

Έστω A ο πίνακας γειτνίασης ενός κατευθυνόμενου γράφου $G = (V, E)$. Να γράψετε έναν αποδοτικό αλγόριθμο (σε μορφή ψευδοκώδικα ή γλώσσας προγραμματισμού C) ο οποίος θα μετασχηματίζει τον A στον πίνακα γειτνίασης που θα αντιστοιχεί στον αντίστροφο γράφο G^T . Ποιο είναι το πλήθος των βασικών πράξεων που απαιτούνται και ποια η ασυμπτωτική του πολυπλοκότητα;

B. (15 μον.)

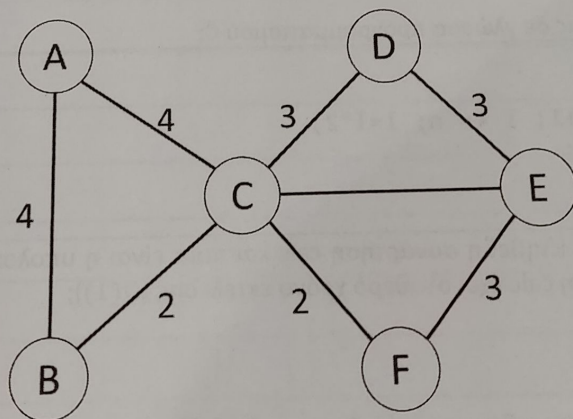
Η παρακάτω αναδρομική συνάρτηση σε γλώσσα προγραμματισμού C, υπολογίζει το διωνυμικό συντελεστή $C(n, k) = \binom{n}{k}$, $n \geq k$, με τη χρήση της ιδιότητας $C(n, k) = C(n-1, k-1) + C(n-1, k)$ και με συνθήκες τερματισμού της αναδρομής $C(n, 0) = C(n, n) = 1$.

```
int binomialCoeff(int n, int k)
{
    if (k > n)
        return 0;
    if (k == 0 || k == n)
        return 1;
    return binomialCoeff(n - 1, k - 1) + binomialCoeff(n - 1, k);
}
```

- Σχεδιάστε το δένδρο αναδρομής για την κλήση της συνάρτησης `binomialCoeff(5, 2)`.
- Υλοποιήστε τον αντίστοιχο αλγόριθμο (σε μορφή ψευδοκώδικα ή γλώσσας προγραμματισμού C) με τη μέθοδο δυναμικού προγραμματισμού για την επίλυση του προβλήματος, χρησιμοποιώντας την αναβιβαστική μέθοδο με υπομνηματισμό (bottom-up) και τη χρήση ενός δισδιάστατου πίνακα. Ποια είναι η υπολογιστική πολυπλοκότητα του αλγορίθμου;

ΘΕΜΑ 4. (25 μον.)

Έστω ο γράφος:

**A. (10 μον.)**

Να υπολογιστεί ο βαθμός της κάθε κορυφής και η λίστα γειτνίασης του γράφου.

B. (15 μον.)

Να υπολογιστεί το ελάχιστο συνδετικό δένδρο με τη χρήση των αλγορίθμων Prim (θεωρώντας ως αρχική κορυφή την A) και Kruskal, δίνοντας το ελάχιστο συνδετικό δένδρο που σχηματίζεται σε κάθε βήμα των δύο αλγορίθμων.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!