ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ «ΠΛΗΡΟΦΟΡΙΚΗ»

ΠΛΗ30: Θεμελιώσεις Επιστήμης Η/Υ

Ακαδημαϊκό έτος 2021 – 22

1η Γραπτή Εργασία

Αλγόριθμοι και Πολυπλοκότητα

Τόμος Α΄, Κεφ. 1 – 4, Kleinberg & Tardos, Κεφ. 5 – 6

Σκοπός της παρούσας γραπτής εργασίας είναι η περαιτέρω εξοικείωση με τον ασυμπτωτικό υπολογισμό μαθηματικών εκφράσεων, τις βασικές έννοιες αναδρομικών και «διαίρει και βασίλευε» αλγορίθμων, και την τεχνική του Δυναμικού Προγραμματισμού.

Η εργασία πρέπει να γραφεί ηλεκτρονικά και να υποβληθεί στον Ψηφιακό Χώρο Εκπαίδευσης του ΕΑΠ (study.eap.gr) σύμφωνα με το Χρονοδιάγραμμα Μελέτης και Γραπτών Εργασιών. Η προθεσμία υποβολής της γραπτής εργασίας είναι η ***Παρασκευή 19 Νοεμβρίου 2021****. Η καταληκτική ημερομηνία υποβολής στο study.eap.gr (συμπεριλαμβανομένης της παράτασης που δίνεται από τον κανονισμό σπουδών) είναι η* ***Τετάρτη 24 Νοεμβρίου 2021****.*

Οδηγίες προς τους φοιτητές:

1. Η υποβολή της εργασίας γίνεται αποκλειστικά μέσω του study.eap.gr και εντός των προκαθορισμένων ημερομηνιών.
2. Υποβάλετε τις απαντήσεις σας χρησιμοποιώντας το «πρότυπο απάντησης» της ΓΕ, που είναι διαθέσιμο στο study.eap.gr, σε επεξεργάσιμη μορφή (.docx) ώστε να δίνεται η δυνατότητα στον Καθηγητή – Σύμβουλο σας να επεξεργαστεί το αρχείο, προσθέτοντας σχόλια και υποδείξεις στα κατάλληλα σημεία.
3. Προσθέστε τις απαντήσεις σας στο χώρο που αντιστοιχεί σε κάθε ερώτημα. Μην αντιγράφετε τις εκφωνήσεις των ερωτημάτων στο αρχείο των απαντήσεων. Μην τροποποιείτε το περιεχόμενο και τη μορφή του αρχείου, προσθέστε μόνο τις απαντήσεις σας στα ερωτήματα. Δεν υπάρχει περιορισμός στον χώρο που θα καταλάβει η απάντηση σας, φροντίστε ωστόσο να είναι συνοπτική, σαφής και πλήρης.
4. Για την ηλεκτρονική υποβολή, για παράδειγμα της 1ης ΓΕ, μετονομάστε το αρχείο σας σύμφωνα με το πρότυπο: Επώνυμο_Όνομα_ΓΕ_1.docx.

ΑΝΑΛΥΤΙΚΟΣ ΠΙΝΑΚΑΣ ΒΑΘΜΟΛΟΓΙΑΣ

Ερώτημα	Μέγιστος βαθμός
1	(3+5+2)+7+8
2	4+(3+7)+9+2
3	(12+5)+8
4	5+7+10+3
Συνολικός Βαθμός:	**100**

Ερώτημα 1

(Α) Έστω f, g μη αρνητικές πραγματικές συναρτήσεις με πεδίο ορισμού το σύνολο των φυσικών αριθμών $\mathbb{N}$.

1. Να αποδείξετε ότι $f(n) + g(n) = \Theta\left(\max\{f(n), g(n)\}\right)$.
2. Να αποδείξετε ότι $f(n) + g(n) = \Omega\left(\min\{f(n), g(n)\}\right)$. Ποια ασυμπτωτική σχέση αρκεί να ισχύει μεταξύ των f, g για να ισχύει επιπλέον $f(n) + g(n) = O\left(\min\{f(n), g(n)\}\right)$; Αιτιολογήστε την απάντησή σας δίνοντας μια κατάλληλη απόδειξη.
3. Αν $0 < g(n) \le f(n) \le g(n) + 1$, για κάθε $n \ge 1$, να αποδείξετε ότι δεν ισχύει αναγκαστικά $f(n) = \Theta(g(n))$, δίνοντας ένα κατάλληλο αντιπαράδειγμα.

(Β) Να αποδείξετε ότι η συνάρτηση $f(n) = 1^1 + 2^2 + \cdots + n^n = \sum_{i=1}^{n} i^i$, με πεδίο ορισμού το σύνολο των φυσικών αριθμών $\mathbb{N}$, ικανοποιεί τη σχέση $f(n) = \Theta(n^n)$.

(Γ) Διατάξτε τις παρακάτω συναρτήσεις, σε αύξουσα σειρά, ως προς τον ρυθμό αύξησής τους:

$f_1(n) = 3^{1.5n}, f_2(n) = (3^n)^2, f_3(n) = n^{10\log_2 n}, f_4(n) = (\log_2 n)^{(\log_2 n)^2}, f_5(n) = n^{100}$.

Ερώτημα 2

Μια ομάδα μηχανικών υπολογιστών που εργάζονται στην εταιρεία DataBite προσπαθεί να εφεύρει νέους αλγόριθμους ανάλυσης μεγάλου όγκου χρηματιστηριακών και τραπεζικών δεδομένων που περιήλθαν στην κατοχή τους. Συγκεκριμένα, έχουν αναπτύξει πέντε αλγορίθμους «διαίρει και βασίλευε» με τα εξής χαρακτηριστικά:

- Ο αλγόριθμος A_1 διασπά το αρχικό πρόβλημα μεγέθους *n* σε τρία υποπροβλήματα μεγέθους *n/3, n/6* και *n/9*, τα επιλύει και στη συνέχεια συνθέτει τις λύσεις τους σε χρόνο *n*.
- Ο αλγόριθμος A_2 διασπά το αρχικό πρόβλημα μεγέθους *n* σε 6 υποπροβλήματα μεγέθους *n/6*, τα επιλύει και στη συνέχεια συνθέτει τις λύσεις τους σε χρόνο 6*n*.
- Ο αλγόριθμος A_3 διασπά το αρχικό πρόβλημα μεγέθους *n* σε 256 υποπροβλήματα μεγέθους *n/4*, τα επιλύει και στη συνέχεια συνθέτει τις λύσεις τους σε χρόνο $8n^4$.
- Ο αλγόριθμος A_4 διασπά το αρχικό πρόβλημα μεγέθους *n* σε 49 υποπροβλήματα μεγέθους *n/343*, τα επιλύει και στη συνέχεια συνθέτει τις λύσεις τους σε χρόνο $5n^2$.
- Ο αλγόριθμος A_5 διασπά το αρχικό πρόβλημα μεγέθους *n* σε 625 υποπροβλήματα μεγέθους *n/125*, τα επιλύει και στη συνέχεια συνθέτει τις λύσεις τους σε χρόνο $10n^{5/4}$.

Προκειμένου να μπορέσουν να εκτιμήσουν τον χρόνο επίλυσης κάθε αλγορίθμου, οι μηχανικοί υπολογιστών ενδιαφέρονται για την ασυμπτωτική πολυπλοκότητα καθενός από αυτούς και ζητούν τη βοήθειά σας. Συγκεκριμένα σας ζητούν:

(Α) Να γράψετε την αναδρομική εξίσωση που δίνει τον χρόνο εκτέλεσης του αλγορίθμου A_1 και στη συνέχεια να την επιλύσετε με το δέντρο της αναδρομής.

(Β) Να γράψετε την αναδρομική εξίσωση που δίνει τον χρόνο εκτέλεσης του αλγορίθμου A_2 και στη συνέχεια:

i. Να την επιλύσετε με χρήση του Θεωρήματος της Κυριαρχίας.
ii. Να επαληθεύστε την απάντηση του υποερωτήματος (i) για τον χρόνο εκτέλεσης $T_2(n)$ του αλγορίθμου A_2 με τη μέθοδο της αντικατάστασης, προσδιορίζοντας τις σταθερές n_0, c_1 και c_2 του ορισμού του ασυμπτωτικού συμβολισμού $\Theta(\ )$ (Τόμος Α', ενότητα 2.1.1, σελ. 35). Ως αρχική συνθήκη, ισχύει ότι $T_2(x) = 1, \forall 0 < x < 1$.

(Γ) Να γράψετε τις αναδρομικές εξισώσεις που δίνουν τον χρόνο εκτέλεσης των αλγορίθμων A_3, A_4 και A_5 και στη συνέχεια να τις επιλύσετε με χρήση του Θεωρήματος της Κυριαρχίας.

(Δ) Να προσδιορίσετε ποιος είναι ο ασυμπτωτικά αποδοτικότερος αλγόριθμος μεταξύ των A_i (i = 1,2,3,4,5) και να δικαιολογήσετε την απάντησή σας.

Ερώτημα 3

Έστω σύνολο S που περιέχει ακέραιους αριθμούς που είναι όλοι διαφορετικοί και έστω $|S| = n$. Θέλουμε να καταχωρίσουμε τα στοιχεία του συνόλου S σε έναν πίνακα $A[0..n-1]$ ώστε να σχηματισθεί δέντρο-σωρός μεγίστων. Ένα *δέντρο-σωρός μεγίστων* είναι ένα πλήρες δυαδικό δέντρο με διατεταγμένους τους κόμβους του έτσι ώστε η τιμή του στοιχείου κάθε κόμβου να είναι μεγαλύτερη από ή ίση με τις τιμές των στοιχείων των παιδιών του (ΠΛΗ10 Τόμος Γ', Ι. Χατζηλυγερούδης, Δομές Δεδομένων, Β' Έκδοση, 2008, σελ. 188). Υπενθυμίζεται ότι σε ένα δέντρο-σωρό αποθηκευμένο σε πίνακα $A[0..m]$, το αριστερό παιδί του κόμβου $A[j]$ αποθηκεύεται στη θέση $A[2j+1]$ (εφόσον $2j+1 \leq m$) και το δεξί παιδί αποθηκεύεται στη θέση $A[2j+2]$ (εφόσον $2j+2 \leq m$).

(Α) (i) Λαμβάνοντας υπόψιν ότι ένα δέντρο-σωρός μεγίστων H αποτελείται από τη ρίζα του (που αποθηκεύει το μέγιστο από τα αποθηκευμένα στοιχεία) και δύο υπο-δέντρα-σωρούς μεγίστων με ρίζες τα παιδιά της ρίζας του H, περιγράψτε αλγόριθμο διαίρει-και-βασίλευε με πολυπλοκότητα χρόνου $O(n \log n)$ για την κατασκευή στον πίνακα $A[0..n-1]$ δέντρου-σωρού μεγίστων με όλα τα στοιχεία του συνόλου S όταν ισχύει ότι $|S| = n = 2^k - 1$ (σε αυτήν την περίπτωση όλα τα επίπεδα του δέντρου-σωρού είναι πλήρη) *χωρίς να χρησιμοποιήσετε ταξινόμηση των στοιχείων οποιουδήποτε υποσυνόλου του συνόλου* S.

Σημείωση: Για να μπορέσετε να καταχωρίσετε τα στοιχεία στη σωστή τους θέση, είναι σκόπιμο σε κάθε αναδρομική κλήση να δίνετε ως όρισμα (επιπλέον του πίνακα A και του προς επεξεργασία υποσυνόλου του συνόλου S) τη θέση (στον πίνακα A) της ρίζας του υπο-δέντρου-σωρού στο οποίο θα αποθηκευτούν τα στοιχεία του υποσυνόλου του S.

(ii) Αποδείξτε ότι η πολυπλοκότητα χρόνου του αλγορίθμου σας είναι $O(n \log n)$.

(Β) Τροποποιήστε τον αλγόριθμό σας για το υποερώτημα Α(i) ώστε να περιγράψετε αλγόριθμο διαίρει-και-βασίλευε με πολυπλοκότητα χρόνου $O(n \log n)$ για την κατασκευή στον πίνακα $A[0..n-1]$ δέντρου-σωρού μεγίστων με όλα τα στοιχεία του συνόλου S για *οποιαδήποτε* τιμή του $|S| = n \geq 1$ *χωρίς να χρησιμοποιήσετε ταξινόμηση των στοιχείων οποιουδήποτε υποσυνόλου του συνόλου* S.

Ερώτημα 4

(Α) Μελετήστε την λύση του προβλήματος του «*σταθμισμένου χρονοπρογραμματισμού διαστημάτων*» στο Κεφ. 6.1 του βιβλίου των Kleinberg & Tardos, και βρείτε τον χρόνο υπολογισμού στην χειρότερη περίπτωση της τιμής του κάθε p(j) με σειριακή αναζήτηση. Υπενθυμίζεται ότι για ένα διάστημα j, η τιμή του p(j) είναι ο μεγαλύτερος δείκτης i < j έτσι ώστε τα διαστήματα i και j να είναι ξένα (δείτε και το βιβλίο).

(Β) Παρατηρήστε ότι στην Εικόνα 6.2 του βιβλίου για την τρέχουσα διάταξη των διαστημάτων, η συνάρτηση p(j) δεν είναι μονότονη καθώς διατρέχουμε τα διαστήματα κατά j= 1, …, 8. Σκεφτείτε πώς θα μπορούσατε να αλλάξετε την τρέχουσα διάταξη ώστε για τη νέα διάταξη β(1), …, β(8) να ισχύει ότι η ακολουθία των τιμών p(β(1)), ..., p(β(8)) είναι αύξουσα.

(Γ) Να σχεδιάσετε έναν αλγόριθμο ο οποίος με είσοδο n διαστήματα διατεταγμένα σε αύξουσα τάξη ως προς το χρόνο λήξης, αρχικά θα βρίσκει τη διάταξη της απάντησης του υποερωτήματος (Β) και στη συνέχεια θα υπολογίζει κατά σειρά τις τιμές p(β(1)), p(β(2)), ... p(β(n)) χρησιμοποιώντας Δυναμικό Προγραμματισμό. Ο αλγόριθμος θα πρέπει να έχει πολυπλοκότητα ασυμπτωτικά μικρότερη από αυτή που βρήκατε για τον αλγόριθμο σειριακής αναζήτησης στο υποερώτημα (Α). *Υπόδειξη:* Βρείτε μια αναδρομική εξίσωση η οποία να εκφράζει το p(β(j)) ως συνάρτηση ενός κατάλληλου πλήθους τιμών p(β(i)), για i < j.

(Δ) Υπολογίστε την ασυμπτωτική χρονική πολυπλοκότητα του αλγορίθμου στο (Γ).