



3.2.2021

Δομές Δεδομένων

Τμήμα Πληροφορικής

Εξεταστική Περίοδος Ιανουαρίου 2021

Λιδάσκων: Ευάγγελος Μαρκάκης

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:	
ΑΜ:	
ΥΠΟΓΡΑΦΗ ΕΠΟΠΤΗ:	

Διάρκεια εξέτασης: 1 ώρα και 40 λεπτά

Απαντήστε σε όλα τα θέματα

Θέμα	Βαθμός
1	
2	
3	
4	
Σύνολο:	

ΘΕΜΑ 1 (20 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 15-20 λεπτά

Απαντήστε στις παρακάτω ερωτήσεις. Δεν χρειάζεται να δικαιολογήσετε την απάντησή σας. Απλώς κυκλώστε τις σωστές απαντήσεις (ενδέχεται σε κάποιες ερωτήσεις να υπάρχει παραπάνω από μία σωστή απάντηση ή ίσως και καμία, εκτός βέβαια αν οι επιλογές είναι Σωστό/Λάθος, όπου υπάρχει ακριβώς μία σωστή απάντηση). Οπουδήποτε παρακάτω δεν εμφανίζεται η βάση κάποιου λογαρίθμου, υποθέστε ότι είναι το 2.

1. Ποιοι από τους παρακάτω ισχυρισμούς είναι σωστοί;
 - a. $8n^6 + 8n^4 - 30n^3 - 100n^2 = O(n^8)$
 - b. $8n^5 + 8n^6 + 30n^3 + 100n^2 = O(n^5)$
 - c. $n^2(\log n)^3 + 3n^3 \log n + 10n^2 = O(n^3(\log n)^2)$
 - d. $(n^2 + n \log n)^2 = O(n^2(\log n)^2)$
 - e. $(\log n)^2 = O(\sqrt{n})$, όπου \sqrt{n} είναι η τετραγωνική ρίζα του n
 - f. $\log_3(n^n) = O(n^{\log n})$
 - g. $(2n+3)^n = O(n^n)$
 - h. $n^3/(\log n)^2 = O(n)$
2. Οι μέθοδοι ταξινόμησης Insertionsort και Heapsort έχουν
 - a. τις ίδιες απαιτήσεις μνήμης
 - b. την ίδια πολυπλοκότητα χειρότερης περίπτωσης
 - c. την ίδια πολυπλοκότητα μέσης περίπτωσης
3. Ένα δέντρο κόκκινου-μαύρου ικανοποιεί τις ιδιότητες που καθορίζονται στον ορισμό των AVL δέντρων.
 - a. Σωστό
 - b. Λάθος
4. Τα τυχαιοποιημένα ΔΔΑ διαφέρουν ασυμπτωτικά από τα στρεβλά ΔΔΑ ως προς
 - a. τον χρόνο αναζήτησης στη μέση περίπτωση
 - b. τον χρόνο αναζήτησης στη χειρότερη περίπτωση
 - c. το ύψος του δέντρου στη χειρότερη περίπτωση από την εισαγωγή n στοιχείων
5. Έστω ένα δέντρο 2-3-4 και ένα AVL δέντρο με τον ίδιο αριθμό αντικειμένων. Έστω h το ύψος του δέντρου 2-3-4. Το ύψος του AVL δέντρου είναι πάντα
 - a. τουλάχιστον h
 - b. τουλάχιστον $2h$
 - c. μικρότερο ή ίσο από h
6. Για την αναδρομική σχέση: $f(n) = f(n/2) + 1$, όταν το n είναι δύναμη του 2 ($n \geq 2$), με $f(1) = 1$, ισχύει ότι $f(n) =$
 - a. $O(\log n)$
 - b. $O(\log \log n)$
 - c. $O(\sqrt{\log n})$

ΘΕΜΑ 2 (15 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 25-30 λεπτά

(α) [10 μονάδες] Δίνεται ο ακόλουθος ορισμός των κόμβων μίας λίστας μονής σύνδεσης.

```
class Node{  
    int key;  
    Node next;  
    Node(int x){ key = x; } }
```

Να γράψετε σε Java τη μέθοδο `Node Fib(int k)` η οποία παίρνει ως είσοδο έναν ακέραιο και δημιουργεί μια κυκλική λίστα, όπου οι κόμβοι της περιέχουν στη σειρά τους πρώτους k αριθμούς Fibonacci. Η ακολουθία των αριθμών Fibonacci δίνεται από την εξίσωση $f_n = f_{n-1} + f_{n-2}$, για $n > 2$, και $f_1 = 1$, $f_2 = 1$. Π.χ. με $k = 6$, η κυκλική λίστα όταν τελειώσει η μέθοδος θα πρέπει να περιέχει 6 κόμβους με τα στοιχεία 1, 1, 2, 3, 5, 8. Η μέθοδος επιστρέφει σύνδεσμο προς τον πρώτο κόμβο (που θα περιέχει τον αριθμό 1) ή null αν $k \leq 0$.

Η μέθοδος πρέπει να τρέχει σε χρόνο $O(k)$, και η έξτρα μνήμη (εκτός από τη μνήμη για την κυκλική λίστα) πρέπει να είναι $O(1)$. Δηλαδή επιτρέπεται να χρησιμοποιήσετε κάποιες βοηθητικές μεταβλητές, αλλά όχι κάποιον πίνακα.

Σημείωση: Δεν πειράζει αν ξεχάσετε κάποιο semicolon ή παρένθεση. Αυτό που προέχει είναι να φαίνονται καθαρά οι εντολές της μεθόδου και η εμβέλεια ενός βρόχου.

(β) [5 μονάδες] Θεωρήστε την εξής μέθοδο, η οποία παίρνει ως είσοδο έναν μη αρνητικό ακέραιο αριθμό:

```
void foo(int n) {  
    if (n==1) System.out.println("Recursion or not?");  
    if (n>1) {  
        for (int i=0; i< n; i++)  
            System.out.println("Recursion or not?");  
        foo(n-1);  
    }  
}
```

Πόσες φορές (ως συνάρτηση του n , με συμβολισμό μεγάλου O) θα τυπώσει η μέθοδος την πρόταση "Recursion or not?"

Για να απαντήσετε την ερώτηση, πρέπει να γράψετε πρώτα μια αναδρομική σχέση που περιγράφει τον αριθμό των γραμμών που τυπώνονται. Κατόπιν, εξηγήστε ποια είναι η λύση της αναδρομικής εξίσωσης.

ΘΕΜΑ 3 (25 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 20-25 λεπτά

(α) [12 μονάδες] Ξεκινώντας από ένα κενό δέντρο δυαδικής αναζήτησης (ΔΔΑ), εκτελέστε εισαγωγή των εξής 8 κλειδιών, με τη σειρά που δίνονται: 29, 13, 9, 5, 31, 11, 20, 24. Εφαρμόστε την απλή εισαγωγή ως φύλλο εκτός από το 3^ο και το 7^ο κλειδί που πρέπει να εισαχθούν με τον αλγόριθμο εισαγωγής στη ρίζα. Σχεδιάστε τα δέντρα που προκύπτουν μετά την ολοκλήρωση της κάθε εισαγωγής (συνολικά 8 δέντρα), και δείξτε επίσης και όλες τις απαιτούμενες περιστροφές που χρειάστηκαν για τα δύο κλειδιά που εισάγονται ως ρίζα (σχεδιάστε δηλαδή και τα ενδιάμεσα δέντρα που προκύπτουν από τις περιστροφές μέχρι τα κλειδιά αυτά να φτάσουν στη ρίζα).

Σημείωση: Αν δεν μπορείτε να κάνετε την εισαγωγή στη ρίζα, εκτελέστε όλες τις εισαγωγές ως φύλλο (με απώλεια τις μισές μονάδες) για να μπορέσετε να έχετε ένα τελικό δέντρο με τα 8 κλειδιά για το επόμενο ερώτημα.

(β) [6 μονάδες] Στο τελικό δέντρο που προέκυψε από το ερώτημα (α), θέλετε να εισάγετε ένα ακόμα κλειδί, το 21. Τώρα όμως θέλετε να τρέξετε τον αλγόριθμο εισαγωγής για τυχαιοποιημένα ΔΔΑ.

Ποια είναι η πιθανότητα το νέο κλειδί να εισαχθεί ως παιδί της ρίζας; Δεν είναι απαραίτητο να σχεδιάσετε κάποιο δέντρο, πρέπει όμως να εξηγήσετε την απάντησή σας (αρκούν 3-5 γραμμές αιτιολόγησης). Μην γράψετε απλά ένα νούμερο, γιατί δεν θα πάρετε καθόλου μονάδες.

(γ) [7 μονάδες] Έστω ένας μεγιστοστρεφής δυαδικός σωρός με ακέραια κλειδιά και με τις εξής ιδιότητες: ο σωρός έχει 7 κόμβους, και περιέχει 3 επίπεδα, το επίπεδο 0 που περιέχει τη ρίζα, το επίπεδο 1 και το επίπεδο 2. Η μεταδιατεταγμένη διάσχιση στο δέντρο επισκέπτεται τελευταίο το κλειδί 39. Στην ενδοδιατεταγμένη διάσχιση, το πρώτο κλειδί που επισκεπτόμαστε είναι το 13 και το προτελευταίο είναι το 27. Τέλος, η προδιατεταγμένη διάσχιση επισκέπτεται τέταρτο το κλειδί 19. Σχεδιάστε ένα σωρό που ικανοποιεί όλες αυτές τις ιδιότητες (υπάρχουν αρκετές πιθανές λύσεις). Αιτιολογήστε συνοπτικά (3-4 γραμμές αρκούν) την απάντησή σας.

ΘΕΜΑ 4 (20 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 20-25 λεπτά

(α) [8 μονάδες] Θέλουμε να αποθηκεύσουμε 6000 αντικείμενα σε πίνακα κατακερματισμού. Έστω ότι υπάρχει αυστηρός περιορισμός ότι το μέσο πλήθος διερευνήσεων για τις ανεπιτυχείς αναζητήσεις δεν πρέπει να υπερβαίνει το 25. Ποιο πρέπει να είναι το ελάχιστο μέγεθος του πίνακα

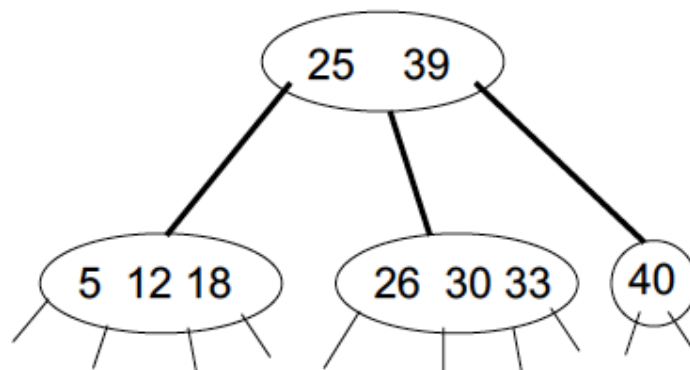
- αν χρησιμοποιήσουμε απλή γραμμική διερεύνηση;
- αν χρησιμοποιήσουμε διπλό κατακερματισμό;

Τι από τα 2 μας συμφέρει περισσότερο, με κριτήριο τις απαιτήσεις μνήμης του κάθε συστήματος;

Το μέσο πλήθος διερευνήσεων στη γραμμική διερεύνηση είναι $(1 + 1/(1-\alpha)^2)/2$ για ανεπιτυχείς αναζητήσεις, όπου α ο συντελεστής φορτίου. Αντίστοιχα, στο διπλό κατακερματισμό είναι $1/(1-\alpha)$.

Σημείωση: Το πρόβλημα μπορεί να απαντηθεί χωρίς να χρειαστεί να λύσετε 2βάθμιες εξισώσεις, και όλες οι εμπλεκόμενες πράξεις μπορούν να γίνουν χωρίς κομπιουτεράκι.

(β) [12 μονάδες] Στο παρακάτω δέντρο 2-3-4, να εισάγετε με ανοδική εισαγωγή πρώτα το κλειδί 15 και μετά το κλειδί 31. Για κάθε μια από τις 2 εισαγωγές, δείξτε τα βήματα που γίνονται καθώς και το δέντρο που προκύπτει όταν ολοκληρώνεται η κάθε εισαγωγή. Σχεδιάστε επίσης και ένα ισοδύναμο δέντρο κόκκινου-μαύρου μόνο για το τελικό δέντρο που προέκυψε μετά τις εισαγωγές.





3.2.2021

Δομές Δεδομένων

Τμήμα Πληροφορικής

Εξεταστική Περίοδος Ιανουαρίου 2021

Διδάσκων: Ευάγγελος Μαρκάκης

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:	
ΑΜ:	
ΥΠΟΓΡΑΦΗ ΕΠΟΠΤΗ:	

Διάρκεια εξέτασης: 1 ώρα και 40 λεπτά

Απαντήστε σε όλα τα θέματα

Θέμα	Βαθμός
1	
2	
3	
4	
Σύνολο:	

ΘΕΜΑ 1 (20 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 15-20 λεπτά

Απαντήστε στις παρακάτω ερωτήσεις. Δεν χρειάζεται να δικαιολογήσετε την απάντησή σας. Απλώς κυκλώστε τις σωστές απαντήσεις (ενδέχεται σε κάποιες ερωτήσεις να υπάρχει παραπάνω από μία σωστή απάντηση ή ίσως και καμία, εκτός βέβαια αν οι επιλογές είναι Σωστό/Λάθος, όπου υπάρχει ακριβώς μία σωστή απάντηση). Οπουδήποτε παρακάτω δεν εμφανίζεται η βάση κάποιου λογαρίθμου, υποθέστε ότι είναι το 2.

1. Ποιοι από τους παρακάτω ισχυρισμούς είναι σωστοί;
 - a. $n^4 \log n + 10n^3 = O(n^3)$
 - b. $8n^5 + 5n^4 + 30n^6 - 100n^2 = O(n^5)$
 - c. $10n^2(\log n)^4 + 3n^3 \log n + 5n^3 + 2n^2 = O(n^3(\log n)^2)$
 - d. $n^4 + 2n^3 + 30n^5 + 9n^2 = O(n^5)$
 - e. $\sqrt{n} = O(n/\log n)$, όπου \sqrt{n} είναι η τετραγωνική ρίζα του n
 - f. $n \log_3(n^7) = O(\log_3 n)$
 - g. $n^n = O(n^{\log n})$
 - h. $n^3/\log n = O(n^2)$
2. Ένας γράφος που έχει N κορυφές, $N-1$ πλευρές και κανένα κύκλο είναι δέντρο.
 - a. Σωστό
 - b. Λάθος
3. Οι μέθοδοι ταξινόμησης Mergesort και Heapsort έχουν
 - a. τις ίδιες απαιτήσεις μνήμης
 - b. την ίδια πολυπλοκότητα χειρότερης περίπτωσης
 - c. την ίδια πολυπλοκότητα μέσης περίπτωσης
4. Τα AVL δέντρα διαφέρουν ασυμπτωτικά από τα δέντρα κόκκινου-μαύρου ως προς
 - a. τον χρόνο αναζήτησης στη μέση περίπτωση
 - b. τον χρόνο αναζήτησης στη χειρότερη περίπτωση
 - c. το ύψος του δέντρου στη χειρότερη περίπτωση από την εισαγωγή n στοιχείων
5. Μπορούμε να κατασκευάσουμε ένα δέντρο 2-3-4 με ύψος 1, που να περιέχει
 - a. 8 αντικείμενα
 - b. 2 αντικείμενα
 - c. 16 αντικείμενα
6. Για την αναδρομική σχέση: $f(n) = 2 + f(n-2)$, όπου n άρτιος, με $f(0) = 1$, ισχύει ότι $f(n) =$
 - a. $O(n)$
 - b. $O(n^2)$
 - c. $O((\log n)^2)$

ΘΕΜΑ 2 (15 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 25-30 λεπτά

(α) [10 μονάδες] Δίνεται η ακόλουθη κλάση για τους κόμβους μίας απλά συνδεδεμένης λίστας, που αποθηκεύει ακέραιους αριθμούς.

```
class Node{  
    int key;  
    Node next;  
}
```

Να γράψετε σε Java τη μέθοδο `Node findMedian(Node h)`, η οποία βρίσκει τον κόμβο που περιέχει τον `median` μιας ταξινομημένης λίστας (σε αύξουσα σειρά) με κεφαλή `h`, τον αφαιρεί από τη λίστα και επιστρέφει την κεφαλή της λίστας που έμεινε. Ο `median` μιας αύξουσας ακολουθίας ακεραίων είναι ο αριθμός που βρίσκεται στη μέση της ακολουθίας. Π.χ., στην ακολουθία 8, 11, 13, 14, 17, 19, 24, ο `median` είναι ο 14. Αν η ακολουθία έχει άρτιο αριθμό ακεραίων, τότε θα κάνουμε τη σύμβαση να παίρνουμε ως `median` τον μεγαλύτερο από τους 2 μεσαίους, δηλαδή στην ακολουθία 8, 11, 13, 17, 19, 24, θεωρούμε ως `median` τον 17.

Αν η αρχική λίστα είναι κενή, η μέθοδος απλά θα τερματίζει επιστρέφοντας `null`. Η μέθοδος πρέπει να τρέχει σε χρόνο $O(N)$, όπου N είναι το πλήθος των κόμβων της λίστας (δεν μπορείτε όμως να υποθέσετε ότι γνωρίζετε εκ των προτέρων πόσο είναι το N).

Σημείωση: Δεν πειράζει αν ξεχάσετε κάποιο `semicolon` ή παρένθεση. Αυτό που προέχει είναι να φαίνονται καθαρά οι εντολές της μεθόδου και η εμβέλεια ενός βρόχου.

(β) [5 μονάδες] Θεωρήστε την εξής μέθοδο, η οποία παίρνει ως είσοδο έναν μη αρνητικό ακέραιο αριθμό:

```
void foo(int n) {  
    if (n==1) System.out.println("Recursion or not?");  
    if (n>1) {  
        for (int i=0; i< n; i++)  
            System.out.println("Recursion or not?");  
        foo(n/2);  
        foo(n/2);  
    }  
}
```

Πόσες φορές (ως συνάρτηση του n , με συμβολισμό μεγάλου O) θα τυπώσει η μέθοδος την πρόταση "Recursion or not?"

Για να απαντήσετε την ερώτηση, πρέπει να γράψετε πρώτα μια αναδρομική σχέση που περιγράφει τον αριθμό των γραμμών που τυπώνονται. Κατόπιν, εξηγήστε ποια είναι η λύση της αναδρομικής εξίσωσης. Για ευκολία, μπορείτε να υποθέσετε ότι η είσοδος n , είναι πάντα δύναμη του 2.

ΘΕΜΑ 3 (25 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 20-25 λεπτά

(α) [12 μονάδες] Ξεκινώντας από ένα κενό δέντρο δυαδικής αναζήτησης (ΔΔΑ), εκτελέστε εισαγωγή των εξής 8 κλειδιών, με τη σειρά που δίνονται: 26, 37, 43, 55, 20, 30, 23, 12. Εφαρμόστε την απλή εισαγωγή ως φύλλο εκτός από το 3^ο και το 7^ο κλειδί που πρέπει να εισαχθούν με τον αλγόριθμο εισαγωγής στη ρίζα. Σχεδιάστε τα δέντρα που προκύπτουν μετά την ολοκλήρωση της κάθε εισαγωγής (συνολικά 8 δέντρα), και δείξτε επίσης και όλες τις απαιτούμενες περιστροφές που χρειάστηκαν για τα δύο κλειδιά που εισάγονται ως ρίζα (σχεδιάστε δηλαδή και τα ενδιάμεσα δέντρα που προκύπτουν από τις περιστροφές μέχρι τα κλειδιά αυτά να φτάσουν στη ρίζα).

Σημείωση: Αν δεν μπορείτε να κάνετε την εισαγωγή στη ρίζα, εκτελέστε όλες τις εισαγωγές ως φύλλο (με απώλεια τις μισές μονάδες) για να μπορέσετε να έχετε ένα τελικό δέντρο με τα 8 κλειδιά για το επόμενο ερώτημα.

(β) [6 μονάδες] Στο τελικό δέντρο που προέκυψε από το ερώτημα (α), θέλετε να εισάγετε ένα ακόμα κλειδί, το 21. Τώρα όμως θέλετε να τρέξετε τον αλγόριθμο εισαγωγής για τυχαιοποιημένα ΔΔΑ.

Ποια είναι η πιθανότητα το νέο κλειδί να εισαχθεί ως παιδί της ρίζας; Δεν είναι απαραίτητο να σχεδιάσετε κάποιο δέντρο, πρέπει όμως να εξηγήσετε την απάντησή σας (αρκούν 3-5 γραμμές αιτιολόγησης). Μην γράψετε απλά ένα νούμερο, γιατί δεν θα πάρετε καθόλου μονάδες.

(γ) [7 μονάδες] Έστω ένας μεγιστοστρεφής δυαδικός σωρός με ακέραια κλειδιά και με τις εξής ιδιότητες: ο σωρός έχει 7 κόμβους, και περιέχει 3 επίπεδα, το επίπεδο 0 που περιέχει τη ρίζα, το επίπεδο 1 και το επίπεδο 2. Η μεταδιατεταγμένη διάσχιση στο δέντρο επισκέπτεται τελευταίο το κλειδί 43. Στην ενδοδιατεταγμένη διάσχιση, το πρώτο κλειδί που επισκεπτόμαστε είναι το 11 και το προτελευταίο είναι το 30. Τέλος, η προδιατεταγμένη διάσχιση επισκέπτεται τέταρτο το κλειδί 21. Σχεδιάστε ένα σωρό που ικανοποιεί όλες αυτές τις ιδιότητες (υπάρχουν αρκετές πιθανές λύσεις). Αιτιολογήστε συνοπτικά (3-4 γραμμές αρκούν) την απάντησή σας.

ΘΕΜΑ 4 (20 μονάδες)

Εκτιμώμενος χρόνος επίλυσης: 20-25 λεπτά

(α) [8 μονάδες] Θέλουμε να αποθηκεύσουμε 4000 αντικείμενα σε πίνακα κατακερματισμού. Έστω ότι υπάρχει αυστηρός περιορισμός ότι το μέσο πλήθος διερευνήσεων για τις ανεπιτυχείς αναζητήσεις δεν πρέπει να υπερβαίνει το 13. Ποιο πρέπει να είναι το ελάχιστο μέγεθος του πίνακα

- αν χρησιμοποιήσουμε απλή γραμμική διερεύνηση;
- αν χρησιμοποιήσουμε διπλό κατακερματισμό;

Τι από τα 2 μας συμφέρει περισσότερο, με κριτήριο τις απαιτήσεις μνήμης του κάθε συστήματος;

Το μέσο πλήθος διερευνήσεων στη γραμμική διερεύνηση είναι $(1 + 1/(1-\alpha)^2)/2$ για ανεπιτυχείς αναζητήσεις, όπου α ο συντελεστής φορτίου. Αντίστοιχα, στο διπλό κατακερματισμό είναι $1/(1-\alpha)$.

Σημείωση: Το πρόβλημα μπορεί να απαντηθεί χωρίς να χρειαστεί να λύσετε 2βάθμιες εξισώσεις, και όλες οι εμπλεκόμενες πράξεις μπορούν να γίνουν χωρίς κομπιουτεράκι.

(β) [12 μονάδες] Στο παρακάτω δέντρο 2-3-4, να εισάγετε με ανοδική εισαγωγή πρώτα το κλειδί 6 και μετά το κλειδί 36. Για κάθε μια από τις 2 εισαγωγές, δείξτε τα βήματα που γίνονται καθώς και το δέντρο που προκύπτει όταν ολοκληρώνεται η κάθε εισαγωγή. Σχεδιάστε επίσης και ένα ισοδύναμο δέντρο κόκκινου-μαύρου μόνο για το τελικό δέντρο που προέκυψε μετά τις εισαγωγές.

