ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
Τ Μ Η Μ Α   Π Λ Η Ρ Ο Φ Ο Ρ Ι Κ Η Σ

09.07.2014

Δομές Δεδομένων

Πτυχιακή Εξεταστική

Ιούλιος 2014

Διδάσκων : Ευάγγελος Μαρκάκης

ΟΝΟΜΑΤΕΠΩΝΥΜΟ:	
ΑΜ:	
ΥΠΟΓΡΑΦΗ ΕΠΟΠΤΗ:	

Διάρκεια εξέτασης : 2 ώρες και 30 λεπτά

Απαντήστε σε όλα τα θέματα

Θέμα	Βαθμός
1	
2	
3	
4	
5	
6	
Σύνολο:	

ΘΕΜΑ 1 **(12 μονάδες)**

Απαντήστε στις παρακάτω ερωτήσεις. Δεν χρειάζεται να δικαιολογήσετε την απάντησή σας. Απλώς κυκλώστε τις σωστές απαντήσεις (ενδέχεται να υπάρχει παραπάνω από μία σωστή απάντηση ή να μην υπάρχει καμία σωστή απάντηση). Δεν υπάρχει αρνητική βαθμολόγηση, για να πάρετε όμως όλες τις μονάδες πρέπει να κυκλώσετε όλες τις σωστές απαντήσεις. Όπου εμφανίζονται λογάριθμοι παρακάτω χωρίς να αναγράφεται ρητά η βάση τους, εννοείται ότι είναι το 2.

1. $8n^5 – 8n^3 + 30n^2 – 100n =$
   a. $O(n^2)$
   b. $O(n^5)$
   c. $O(n^8)$
   d. $O(n!)$

2. Οι μέθοδοι ταξινόμησης Quicksort και Insertionsort έχουν
   a. τις ίδιες απαιτήσεις μνήμης
   b. την ίδια πολυπλοκότητα χειρότερης περίπτωσης
   c. την ίδια πολυπλοκότητα μέσης περίπτωσης

3. Σε ένα B-δέντρο με παράμετρο M, όπου M=O(1), και με n αντικείμενα, ο χρόνος αναζήτησης στη χειρότερη περίπτωση είναι
   a. $O(\log n)$
   b. $O((\log_M n)^5)$
   c. $O(\log_4(\log_M n))$
   d. $O(\mathrm{sqrt}(\log n))$

4. Το ύψος ενός δέντρου δυαδικής αναζήτησης με n κόμβους, είναι πάντα
   a. τουλάχιστον $\log n – 2$
   b. τουλάχιστον $\log n + 2$
   c. το πολύ $n-2$
   d. το πολύ $n+2$

5. Η συνάρτηση $n^{\log n}$ είναι
   a. $O(n!)$
   b. $O(n^{10})$
   c. $O(n^n)$
   d. $O(\log n)$

6. Έστω η αναδρομική σχέση $f(n) = f(n-1) + n$, για $n \geq 2$, και $f(1) = 1$. Τότε $f(n) =$
   a. $O(n\log n)$
   b. $O(n)$
   c. $O(n^2)$
   d. $O(n(\log n)^2)$

Απαντήσεις:

1. **b, c, d**

2. **a, b**
3. **a, b**
4. **a, d**
5. **a, c**
6. **c**

ΘΕΜΑ 2 (12 μονάδες)

Δίνεται ο ακόλουθος ορισμός των κόμβων μίας απλά συνδεδεμένης λίστας. Κάθε κόμβος περιέχει ένα ακέραιο κλειδί key και ένα δείκτη next προς κάποιον άλλο κόμβο (ή είναι null).

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

Να γράψετε σε Java τη μέθοδο `Node insert(Node h, int x)` η οποία υλοποιεί την εισαγωγή ενός κόμβου με κλειδί x στην *ταξινομημένη* λίστα μονής σύνδεσης με κεφαλή h. Η λίστα είναι ταξινομημένη σε αύξουσα σειρά και πρέπει να διατηρείται ταξινομημένη μετά την εισαγωγή. Αν το κλειδί υπάρχει ήδη στη λίστα, δεν γίνεται εισαγωγή και η μέθοδος επιστρέφει null, αλλιώς επιστρέφει τον κόμβο που εισήγαγε. Η λίστα δεν έχει κόμβους φρουρούς και τερματίζεται με null.

Υπάρχουν πάρα πολλοί τρόποι για να υλοποιηθεί μια τέτοια μέθοδος. Ενδεικτικά:

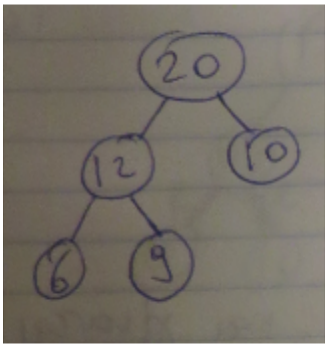
```
Node insert(Node h, int x)
{
      if (h==null) return new Node(x); //μην ξεχνάτε τον έλεγχο αυτό!!
      if (x==h.key) return null;
      if (x<h.key)
            { //εισαγωγή στην αρχή, την αντιμετωπίζουμε χωριστά
                  Node t = new Node(x);
                  t.next = h;
                  h=t;
                  return h;
            }
     //αν φτάσουμε εδώ η εισαγωγή πρέπει να γίνει μετά την κεφαλή
     Node t = h;
     while (t.next!=null)
            {
                  if (x==t.next.key) return null; //αν υπάρχει επιστρέφει null
                  if (x < t.next.key) //βρήκαμε πού πρέπει να μπει
                        {
                              Node s = new Node(x);
                              s.next = t.next;
                              t.next = s;
                              return s;
                        }
                  else t = t.next;
            } // end of while
      Node s = new Node(x); //διαφορετικά, εισαγωγή στο τέλος
      t.next = s;
      return s; }
```

ΘΕΜΑ 3 (15 μονάδες)

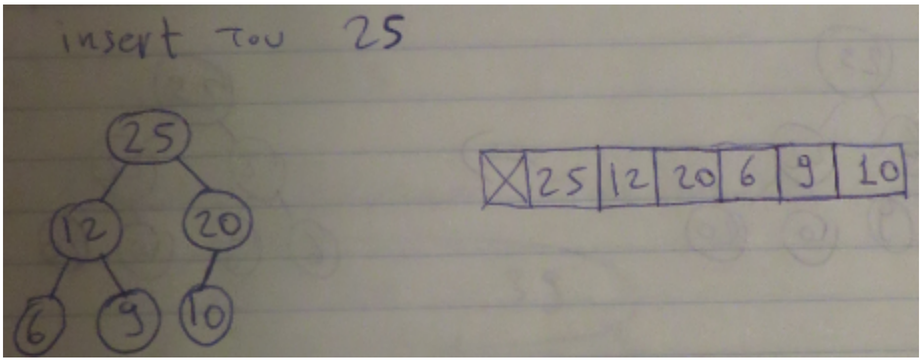
Έστω ότι έχουμε ένα σωρό που υποστηρίζει τις βασικές λειτουργίες `insert` και `getmax`, όπως τις είδαμε στο μάθημα. Ο σωρός αρχικά έχει την ακόλουθη μορφή (το στοιχείο στη θέση 0 του πίνακα δεν χρησιμοποιείται, για αυτό και το στοιχείο 20 είναι στη θέση 1):

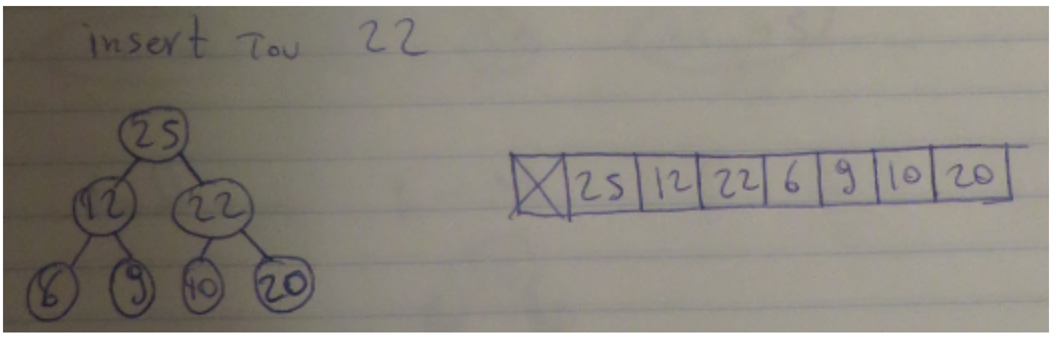
	20	12	10	6	9

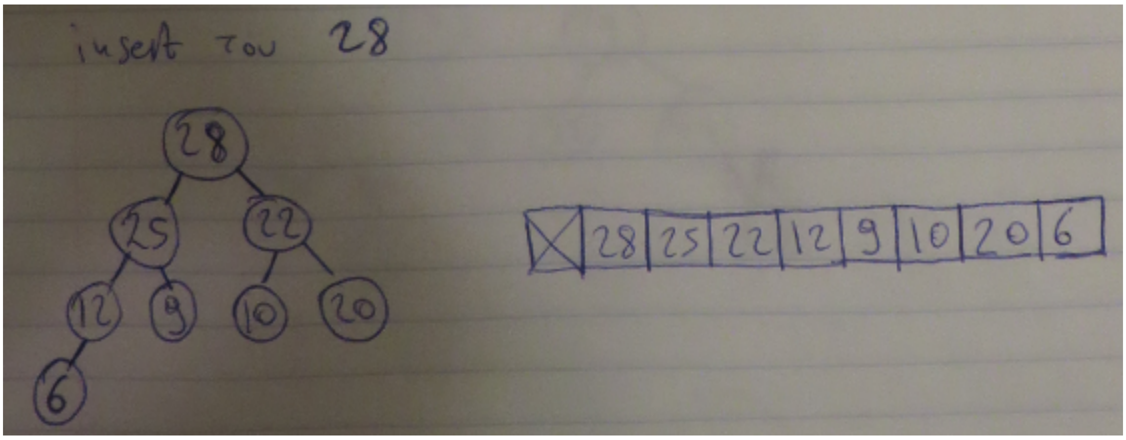
α) Σχεδιάστε την αναπαράσταση του σωρού ως πλήρες δυαδικό δέντρο.



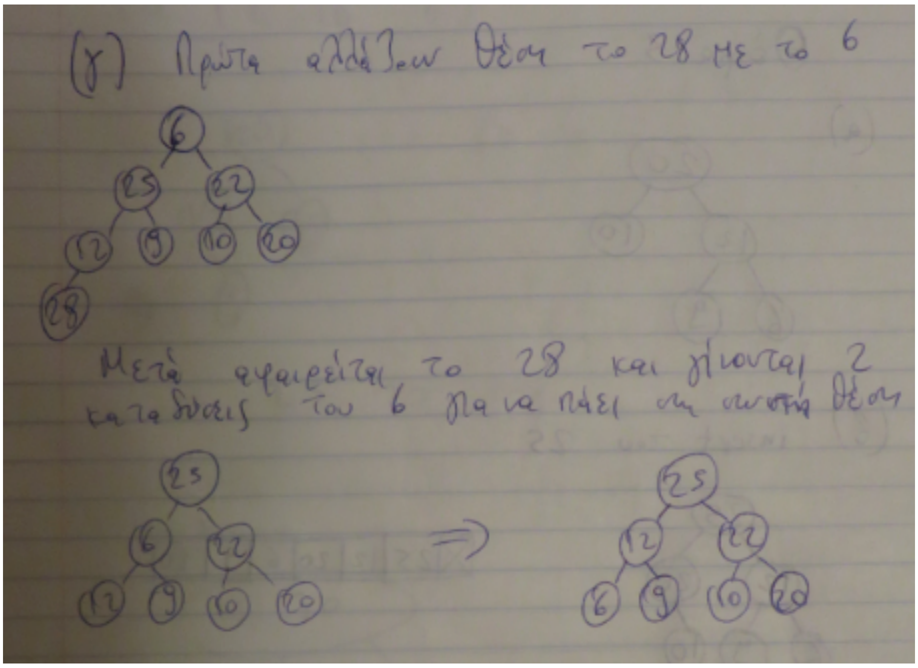
β) Έστω ότι ξεκινώντας από το σωρό του ερωτήματος (α), εισάγουμε διαδοχικά τα στοιχεία 25, 22, 28, σύμφωνα με τη μέθοδο `insert`, η οποία όπως έχουμε δει, αρχικά εισάγει ένα στοιχείο στο τέλος και στη συνέχεια καλεί την μέθοδο της ανάδυσης (`swim`) για να αποκαταστήσει την ιδιότητα του σωρού. Να σχεδιάσετε τον πίνακα και το αντίστοιχο πλήρες δυαδικό δέντρο που προκύπτει για τον σωρό, όταν ολοκληρώνεται κάθε κλήση της `insert`.







γ) Εκτελέστε μία αφαίρεση μεγίστου (`getmax`) στο σωρό που έχει προκύψει από το ερώτημα (β). Δείξτε όλες τις αλλαγές που γίνονται (μετακινήσεις στοιχείων) μέχρι να ολοκληρωθεί η κλήση της `getmax`.



ΘΕΜΑ 4 (12 μονάδες)

(α) [8 μονάδες] Έστω ότι εφαρμόζετε κατακερματισμό με γραμμική διερεύνηση σε 6,000 αντικείμενα, και έχετε χρησιμοποιήσει πίνακα μεγέθους 12,000. Έστω ότι για τον ίδιο αριθμό αντικειμένων θέλετε να έχετε και ένα σύστημα που χρησιμοποιεί χωριστή αλυσίδωση και έχει τον ίδιο μέσο όρο διερευνήσεων για **ανεπιτυχείς αναζητήσεις** με το προηγούμενο. Ποιο θα πρέπει να είναι το μέγεθος του πίνακα στη χωριστή αλυσίδωση; Για ανεπιτυχείς αναζητήσεις, το μέσο πλήθος διερευνήσεων στη γραμμική διερεύνηση είναι $(1 + 1/(1-\alpha)^2)/2$, όπου α ο συντελεστής φορτίου.

(β) [4 μονάδες] Να συγκρίνετε το μέσο πλήθος διερευνήσεων για **επιτυχείς αναζητήσεις** στα 2 παραπάνω συστήματα. Για επιτυχείς αναζητήσεις, το μέσο πλήθος διερευνήσεων είναι $(1 + 1/(1-\alpha))/2$, στη γραμμική διερεύνηση.

(α)

Έχουμε N = 6000. Για το σύστημα της γραμμ. διερεύνησης έχουμε ότι $\alpha_{\gamma\delta} = ½$. Αυτό σημαίνει ότι με βάση τον τύπο, ο αριθμός διερευνήσεων για ανεπιτυχείς αναζητήσεις στο σύστημα γραμμ. διερεύνησης είναι $(1 + 1/(1-1/2)^2)/2 = 5/2$.

Για να έχουμε τώρα ένα ισοδύναμο σύστημα χωριστής αλυσίδωσης, ως προς τις διερευνήσεις για ανεπιτυχείς αναζητήσεις, έστω $M_{\chi\alpha}$ το μέγεθος του πίνακα που θα χρησιμοποιήσουμε στο σύστημα αυτό. Ο μέσος όρος διερευνήσεων για ανεπιτυχείς αναζητήσεις στη χωριστή αλυσίδωση είναι $N/M_{\chi\alpha}$. Άρα πρέπει να ισχύει ότι $N/M_{\chi\alpha} = 5/2$. Από αυτή τη σχέση προκύπτει ότι $M_{\chi\alpha} = 2400$. Άρα το μέγεθος του πίνακα πρέπει να είναι 2400.

(β)

Ο αριθμός διερευνήσεων για επιτυχείς αναζητήσεις στη χωριστή αλυσίδωση είναι $N/(2M_{\chi\alpha})$, καθώς κατά μέσο όρο στις επιτυχείς αναζητήσεις, μέχρι τη μέση της λίστας έχουμε βρει το κλειδί που ψάχνουμε. Άρα στη χωριστή αλυσίδωση είναι 5/4.
Για το σύστημα γραμμικής διερεύνησης, εφαρμόζοντας τον τύπο παίρνουμε $(1 + 1/(1-1/2))/2 = 3/2$.
Άρα με βάση το κριτήριο του αριθμού διερευνήσεων για επιτυχείς αναζητήσεις, η χωριστή αλυσίδωση είναι καλύτερη.

ΘΕΜΑ 5 (17 μονάδες)

(α) [5 μονάδες] Θεωρήστε ένα ισορροπημένο δέντρο 2-3-4 με n κλειδιά (ισορροπημένο σημαίνει ότι όλοι οι σύνδεσμοι ισαπέχουν από τη ρίζα). Έστω ότι το μετατρέπετε στην ισοδύναμη αναπαράσταση σε δέντρο κόκκινου-μαύρου. Πόσο αυξάνεται το ύψος του δέντρου στη χειρότερη περίπτωση κατά τη μετατροπή αυτή σε σχέση με το αρχικό δέντρο 2-3-4; Περιγράψτε από πού προέρχεται η αύξηση αυτή στο ύψος του δέντρου. Μην χρησιμοποιήσετε κομμάτια κώδικα για να απαντήσετε. Περιγράψτε απλά ποια είναι τα βήματα που μπορεί να προκαλέσουν την αύξηση αυτή, και τι συνεπάγεται αυτό για το ύψος του τελικού δέντρου κόκκινου-μαύρου στη χειρότερη περίπτωση.

Απ: Στη χειρότερη περίπτωση το ύψος του δέντρου διπλασιάζεται. Η αύξηση του ύψους προκαλείται από τη μετατροπή των 3-κόμβων και 4-κόμβων σε απλούς κόμβους. Συγκεκριμένα, έχουμε δει ότι η μετατροπή από 2-3-4 σε κόκκινο-μαύρο έχει ως αποτέλεσμα την αντικατάσταση ενός 3-κόμβου ή 4-κόμβου από υποδέντρα με 2 επίπεδα. Συνεπώς το ύψος μπορεί να αυξηθεί κατά 1 για κάθε 3-κόμβο ή 4-κόμβο. Στη χειρότερη περιπτωση λοιπόν, αν έχουμε 3-κόμβους ή 4-κόμβους σε κάθε επίπεδο του δέντρου, το ύψος μπορεί μέχρι και να διπλασιαστεί.

(β) [8 μονάδες] Διατάξτε τις ακόλουθες συναρτήσεις με βάση τον αυξητικό τους χαρακτήρα, δηλαδή βρείτε μια διάταξη $f_1, \ldots, f_8$ των οκτώ συναρτήσεων παρακάτω, η οποία να ικανοποιεί τις σχέσεις $f_1 = O(f_2)$, $f_2 = O(f_3)$, …, κ.ο.κ. (η λύση μπορεί να μην είναι μοναδική, εσείς απλά γράψτε μία διάταξη που να ικανοποιεί τις σχέσεις αυτές). Σε όσους λογαρίθμους παρακάτω δεν εμφανίζεται η βάση τους, θεωρήστε πως είναι ίση με 2.

n^3 $n^{1/4}$ $\log(n^4)$ $n^3\log n$ n^n $\log_4(n^4)$ $4^{n\log n}$ $n^2\mathrm{sqrt}(n)$

Απ:

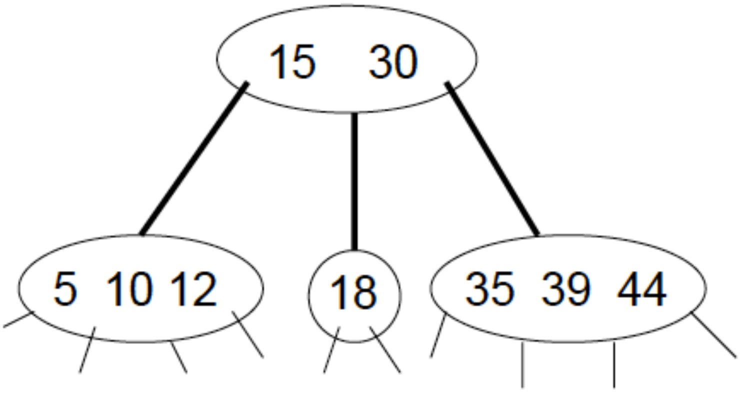
1. $\log(n^4)$ ή $\log_4(n^4)$
2. $\log_4(n^4)$ ή $\log(n^4)$
3. $n^{1/4}$
4. $n^2\mathrm{sqrt}(n)$
5. n^3
6. $n^3\log n$
7. n^n
8. $4^{n\log n}$

(γ) [4 μονάδες] Γιατί στα Β-δέντρα χρησιμοποιούμε κόμβους που περιέχουν σχετικά μεγάλο αριθμό κλειδιών;

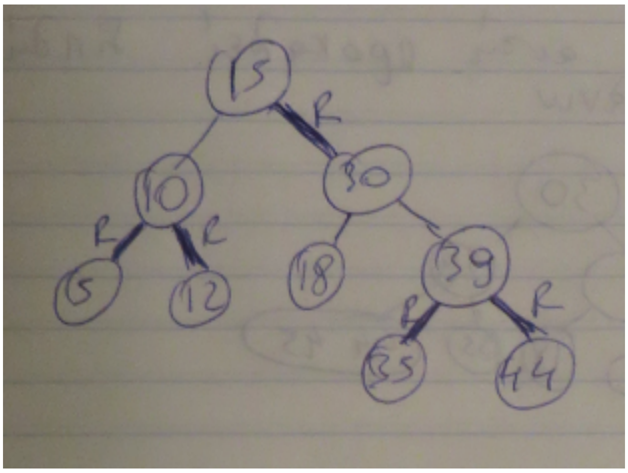
Απ: Γιατί τα Β-δεντρα τα χρησιμοποιούμε συνήθως για εξωτερική αναζήτηση, όπου το κύριο κόστος είναι το πόσες φορές διαβάζουμε/γράφουμε στο σκληρό δίσκο ή στη βάση δεδομένων. Προσπαθούμε λοιπόν κάθε κόμβος να περιέχει περίπου όσα κλειδιά χωράνε σε μια σελίδα/μπλοκ του δίσκου, (η ανάγνωση από το δίσκο ούτως ή άλλως ένα μπλοκ θα επιστρέψει και όχι μεμονωμένα κλειδιά) και να ελαχιστοποιούμε έτσι το πόσες φορές θα διαβάσουμε από το δίσκο.

ΘΕΜΑ 6 **(12 μονάδες)**

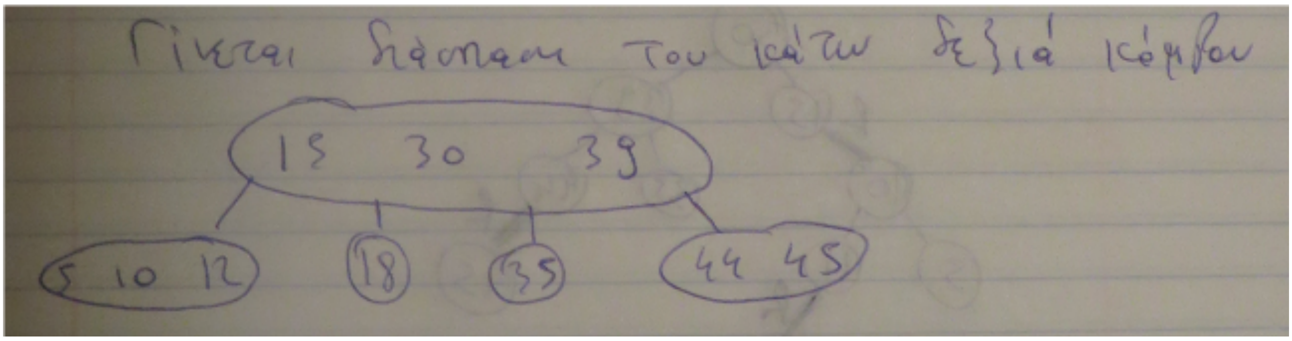
Έστω το εξής δέντρο 2-3-4:

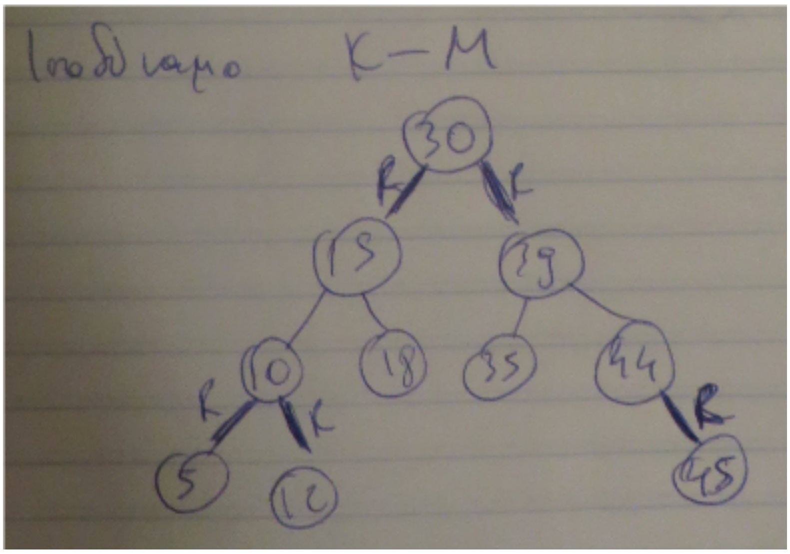


(α) Σχεδιάστε ένα ισοδύναμο δέντρο κόκκινου-μαύρου.



(β) Να εισάγετε με ανοδική εισαγωγή το κλειδί 45. Σχεδιάστε το νέο δέντρο 2-3-4 που θα προκύψει καθώς και το ισοδύναμο δέντρο κόκκινου-μαύρου.





(γ) Στο δέντρο που προέκυψε από το ερώτημα (β) κάντε ανοδική εισαγωγή του κλειδιού 11. Ξανασχεδιάστε το νέο δέντρο 2-3-4 και το ισοδύναμο κόκκινου-μαύρου.