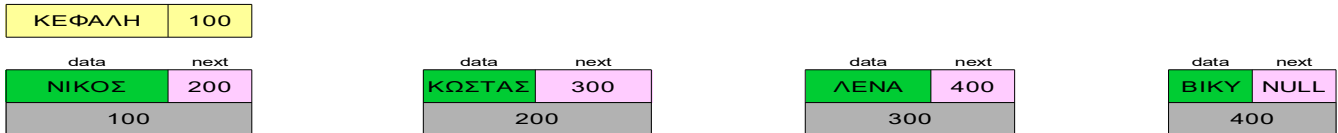


Ασκήσεις στις Δυναμικές δομές δεδομένων

Σημείωση: Στις επόμενες ασκήσεις κάθε κόμβος απεικονίζεται σε δύο γραμμές. Στην πάνω γραμμή έχουμε τα δεδομένα του κόμβου (**data**) και τις διευθύνσεις μνήμης των κόμβων με τους οποίους συνδέεται ο απεικονιζόμενος κόμβος (π.χ. **next**, **prev**, **left**, **right**, **middle**). Στην κάτω γραμμή έχουμε τη διεύθυνση (θέση) στη μνήμη (RAM) του απεικονιζόμενου κόμβου. Για την προσπέλαση των κόμβων χρειαζόμαστε την ειδική μεταβλητή-δείκτη: **ΚΕΦΑΛΗ** που περιέχει την διεύθυνση (θέση στη μνήμη) του πρώτου προσπελάσιμου κόμβου της δομής. Στις διπλά συνδεδεμένες λίστες χρησιμοποιείται επιπλέον και η ειδική μεταβλητή-δείκτης **ΟΥΡΑ**.

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ



1. Δίνεται η παραπάνω απεικόνιση μιας δυναμικής δομής δεδομένων στη μνήμη ενός υπολογιστή που έχει τη μορφή μιας απλά συνδεδεμένης λίστας, με 4 χαϊδευτικά ονόματα.

α) Υποθέστε ότι η λίστα αυτή υλοποιεί μια **Δυναμική στοίβα** (θεωρήστε ότι ο δείκτης **ΚΕΦΑΛΗ** είναι ο δείκτης **top** και δείχνει τον κόμβο που ωθήθηκε τελευταίος στην λίστα-στοίβα)

- I. Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **ώθηση** ενός νέου κόμβου στη στοίβα, με τιμή ΤΑΣΟΣ που βρίσκεται αποθηκευμένος στη θέση μνήμης 500.
- II. Να αναπαραστήσετε τη λίστα μετά την **απώθηση** ενός κόμβου απ' την αρχική στοίβα.

β) Υποθέστε ότι η λίστα αυτή υλοποιεί μια **Δυναμική ουρά** (θεωρήστε ότι ο δείκτης **ΚΕΦΑΛΗ** είναι ο δείκτης **rear** της ουράς και δείχνει τον κόμβο που εισήχθη τελευταίος στην λίστα-ουρά, ενώ η τιμή του δείκτη **front** είναι η θέση μνήμης του αρχικά εισαγόμενου κόμβου.

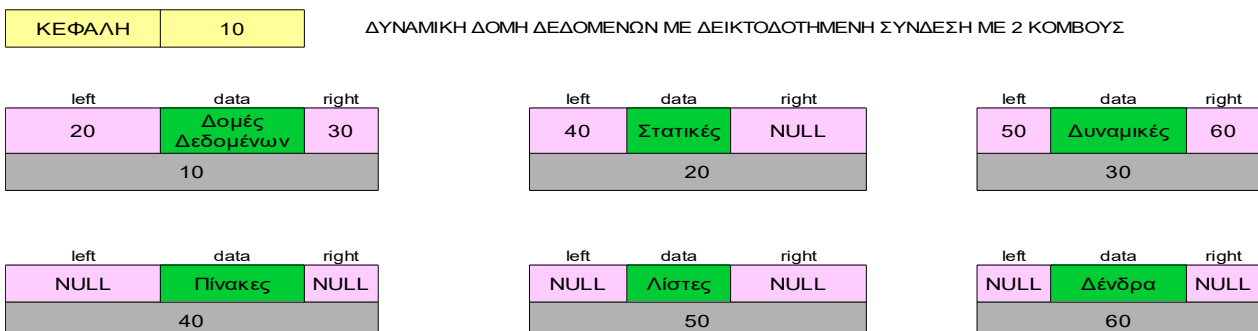
(Εδώ, το πρώτο όνομα που εισήχθη είναι το ΒΙΚΥ και συνεπώς η τιμή του front είναι 400, αλλά δεν υπάρχει δυνατότητα άμεσης προσπέλασης σ' αυτόν (δες την άσκηση 7, σε υλοποίηση με διπλά συνδεδεμένη λίστα))

- I. Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **εισαγωγή** ενός νέου κόμβου στην ουρά με τιμή ΤΑΣΟΣ που βρίσκεται αποθηκευμένος στη θέση μνήμης 450.
- II. Να αναπαραστήσετε τη λίστα μετά την **εξαγωγή** ενός κόμβου απ' την αρχική ουρά.

γ) Υποθέστε ότι η δομή αυτή υλοποιεί μια **Δυναμική απλά συνδεδεμένη λίστα**.

- I. Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **εισαγωγή** στη θέση μετά τον κόμβο με τιμή ΛΕΝΑ, ενός νέου κόμβου με τιμή ΤΑΣΟΣ που βρίσκεται αποθηκευμένος στη θέση μνήμης 450.
- II. Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **διαγραφή** του κόμβου με τιμή ΛΕΝΑ απ' την αρχική λίστα.

2. Δίνεται η παρακάτω απεικόνιση μιας δυναμικής δομής δεδομένων στη μνήμη ενός υπολογιστή με δύο δείκτες (pointers) σε κάθε κόμβο της, όπου ο καθένας δείχνει αντίστοιχα σ' ένα παιδί του κόμβου. Η δομή αυτή υλοποιεί ένα **δυναμικό δένδρο**.



α) Να σχεδιάσετε τη δομή του δένδρου.

β) Αν εισαχθούν στο δένδρο δύο επιπλέον κόμβοι, ως παιδιά του κόμβου με τιμή **Λίστες**, με τιμές: **Απλές Λίστες**, για το αριστερό παιδί και **Διπλές Λίστες** για το δεξί παιδί, που βρίσκονται αποθηκευμένοι στις θέσεις μνήμης 70, 80 αντίστοιχα τότε:

- Να ξανασχεδιάσετε τη δενδρική δομή ώστε να περιέχει και τους 2 νέους κόμβους
- Να αναπαράσχετε τη νέα δομή στη μνήμη ώστε να υλοποιεί το νέο δένδρο.

3. Δίνεται η παρακάτω απεικόνιση μιας δυναμικής δομής δεδομένων στη μνήμη ενός υπολογιστή με τρεις δείκτες (pointers) σε κάθε κόμβο της. Ο καθένας δείχνει αντίστοιχα σε ένα παιδί του κόμβου. Αυτό σημαίνει ότι κάθε κόμβος μπορεί να έχει από κανένα, μέχρι και 3 παιδιά.

ΔΥΝΑΜΙΚΗ ΔΟΜΗ ΔΕΔΟΜΕΝΩΝ ΜΕ ΔΕΙΚΤΟΔΟΤΗΜΕΝΗ ΣΥΝΔΕΣΗ ΜΕ 3 ΚΟΜΒΟΥΣ

ΚΕΦΑΛΗ	10
--------	----

data	left	middle	right
Δομές Δεδομένων	20	NULL	30
10			

data	left	middle	right
Στατικές	40	90	100
20			

data	left	middle	right
Δυναμικές	110	NULL	120
30			

data	left	middle	right
Πίνακες	NULL	NULL	NULL
40			

data	left	middle	right
Λίστες	70	NULL	80
50			

data	left	middle	right
Γραμμικές	90	100	50
110			

data	left	middle	right
Στοίβα	NULL	NULL	NULL
90			

data	left	middle	right
Απλής Σύνδεσης	NULL	NULL	NULL
70			

data	left	middle	right
Μη Γραμμικές	60	NULL	130
120			

data	left	middle	right
Ουρά	NULL	NULL	NULL
100			

data	left	middle	right
Διπλής Σύνδεσης	NULL	NULL	NULL
80			

data	left	middle	right
Δένδρα	140	NULL	NULL
60			

data	left	middle	right
Διαδικά Δένδρα Αναζήτησης	NULL	NULL	NULL
140			

data	left	middle	right
Γράφοι	150	NULL	160
130			

data	left	middle	right
Κατευθυνόμενοι Γράφοι	NULL	NULL	NULL
150			

data	left	middle	right
Μη Κατευθυνόμενοι Γράφοι	NULL	NULL	NULL
160			

- α) Να αναπαραστήσετε γραφικά τους κόμβους και τις συνδέσεις αυτής της δομής δεδομένων.
β) Να προσδιορίσετε τον τύπο της δομής, δηλαδή αν αποτελεί **Δένδρο ή Γράφο**.

4. Δίνεται η παρακάτω απεικόνιση μιας δενδρικής δυναμικής δομής δεδομένων στη μνήμη ενός υπολογιστή με δύο δείκτες (pointers) σε κάθε κόμβο της, η οποία υλοποιεί ένα δυαδικό δένδρο που περιέχει ονόματα νησιών των Κυκλάδων. Ο πρώτος δείκτης δείχνει το αριστερό παιδί και ο δεύτερος το δεξί.

ΔΥΝΑΜΙΚΗ ΔΟΜΗ ΜΕ 2 ΔΕΙΚΤΕΣ ΣΕ ΚΑΘΕ ΚΟΜΒΟ



- α) Να σχεδιάσετε το δένδρο σύμφωνα με τις συνδέσεις που υποδεικνύει η δομή στη μνήμη.
β) Να εξετάσετε αν αποτελεί δυαδικό δένδρο αναζήτησης. Αν δεν είναι, να το τροποποιήσετε ώστε να γίνει (να το επανασχεδιάσετε) και να διορθώσετε αντίστοιχα και την δεικτοδότηση της δομής στη μνήμη (να την αναπαραστήσετε ξανά).
γ) Να εισαγάγετε στο δένδρο 3 νέους κόμβους με τα νησιά Αμοργός, Ανάφη, Ίος που βρίσκονται στις θέσεις μνήμης 160,170,180 αντίστοιχα και να αλλάξετε το όνομα του κόμβου Θήρα σε Σαντορίνη. Μετά τις αλλαγές αυτές το δένδρο πρέπει να παραμείνει δυαδικό δένδρο αναζήτησης. Να σχεδιάσετε το νέο δένδρο.
δ) Να αναπαραστήσετε τη νέα δυναμική δομή στη μνήμη ώστε να υλοποιεί το νέο δένδρο.
ε) Πόσες συγκρίσεις απαιτούνται για την εύρεση του κόμβου 'Πάρος' στο δένδρο;

5. Δίνεται ο παρακάτω ταξινομημένος πίνακας που περιέχει 15 ονόματα νησιών των Κυκλάδων:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ΑΜΟΡΓΟΣ	ΑΝΔΡΟΣ	ΙΟΣ	ΚΕΑ	ΚΥΘΝΟΣ	ΜΗΛΟΣ	ΜΥΚΟΝΟΣ	ΝΑΞΟΣ	ΠΑΡΟΣ	ΣΑΝΤΟΡΙΝΗ	ΣΕΡΙΦΟΣ	ΣΙΚΙΝΟΣ	ΣΙΦΙΟΣ	ΣΥΡΟΣ	ΤΗΝΟΣ

α) Να μετατρέψετε τον πίνακα αυτόν σε δυαδικό δένδρο αναζήτησης. Να σχεδιάσετε το δένδρο αυτό λαμβάνοντας υπόψη τον αλγόριθμο της Δυαδικής αναζήτησης (μέθοδος Διάρει και Βασίλειου). Υπενθυμίζεται ότι η μέθοδος αυτή στηρίζεται στην διαδοχική εύρεση του μεσαίου στοιχείου ενός πίνακα ή υποπίνακα, και στη διαδοχική διχοτόμηση του πίνακα ή του υποπίνακα αυτού, με κέντρο το μεσαίο στοιχείο, σε δύο νέους υποπίνακες κάθε φορά.

Έτσι η σχεδίαση του αντίστοιχου δένδρου γίνεται ακολουθώντας τα επόμενα βήματα:

Το στοιχείο της μεσαίας θέσης του αρχικού πίνακα αποτελεί την ρίζα του ζητούμενου προς σχεδίαση δένδρου.

- Το μεσαίο στοιχείο του αριστερού υποπίνακα αποτελεί το αριστερό παιδί της ρίζας του αντίστοιχου σχεδιαζόμενου υποδένδρου.
- Το μεσαίο στοιχείο του δεξιού υποπίνακα αποτελεί το δεξί παιδί της ρίζας του αντίστοιχου σχεδιαζόμενου υποδένδρου.

Το μεσαίο στοιχείο του υποπίνακα που εξετάζεται κάθε φορά, είναι ο νέος κόμβος που προστίθεται στο υπό σχεδίαση ζητούμενο δένδρο.

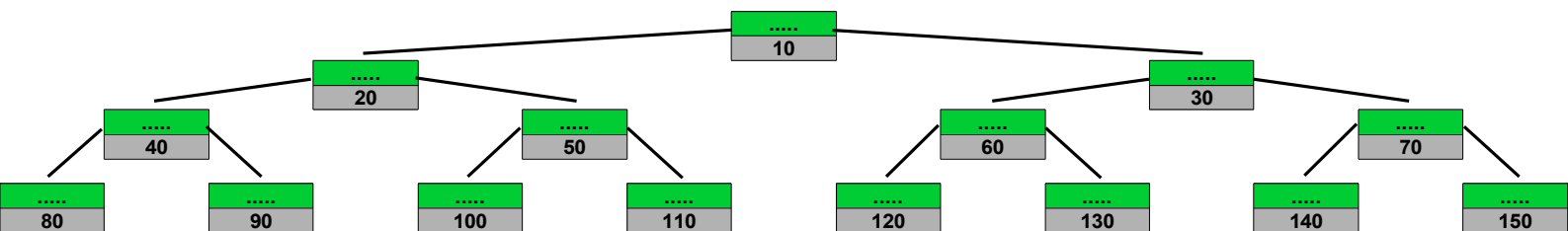
Επαναλαμβάνουμε διαδοχικά τα βήματα (i και ii) διχοτομώντας κάθε διχοτομημένο υποπίνακα, εφ' όσον βέβαια μπορεί να διχοτομηθεί κι' άλλο, μέχρι να προστεθούν στο δένδρο όλα τα στοιχεία του αρχικού πίνακα.

β) Πόσοι έλεγχοι απαιτούνται στη χειρότερη περίπτωση (οι περισσότεροι δυνατοί έλεγχοι) για να βρεθεί οποιοδήποτε νησί στο δένδρο που σχεδιάσατε;

Απαιτούνται οι ίδιοι και στην δυαδική αναζήτηση οποιουδήποτε νησιού στον ταξινομημένο πίνακα;

6. Δίνεται το παρακάτω δυαδικό δένδρο αναζήτησης με 15 κόμβους. Η τιμή κάθε κόμβου αποκρύβεται και εμφανίζεται μόνο η θέση μνήμης στην οποία είναι καταχωρημένη. Για παράδειγμα η τιμή του αριστερότερου φύλλου είναι καταχωρημένη στη θέση μνήμης 80. Γνωρίζουμε ότι κάθε δυαδικό δένδρο αναζήτησης μπορεί να μετατραπεί σε ένα ταξινομημένο πίνακα και αντίστροφα.

Να μετατρέψετε το δένδρο σε αντίστοιχο πίνακα 15 θέσεων, **ταξινομημένο ως προς τις άγνωστες τιμές των κόμβων του δένδρου**. Να σχεδιάσετε αυτόν τον πίνακα όπως φαίνεται παρακάτω και σε κάθε θέση του να τοποθετήσετε αντί της άγνωστης τιμής του κόμβου, την αντίστοιχη θέση που κατέχει αυτή στη μνήμη.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

7. Δίνεται η παρακάτω απεικόνιση μιας δυναμικής δομής δεδομένων στη μνήμη ενός υπολογιστή που έχει τη μορφή μιας διπλά συνδεδεμένης λίστας. Κάθε κόμβος διαθέτει δύο δείκτες. Ο next δείχνει τον επόμενο κόμβο και ο prev (previous) τον προηγούμενο.

α) Υποθέστε ότι η λίστα αυτή υλοποιεί μια **Δυναμική ουρά**. Υποθέστε ότι η ειδική μεταβλητή-δείκτης **ΚΕΦΑΛΗ** αποτελεί τον **εμπρός (front)** δείκτη της ουράς και η μεταβλητή-δείκτης **ΟΥΡΑ** αποτελεί τον **πίσω (rear)** δείκτη.

I) Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **εισαγωγή** ενός νέου κόμβου στην δυναμική ουρά με τιμή ΤΑΣΟΣ που βρίσκεται αποθηκευμένη στη θέση μνήμης 450.

II) Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **εξαγωγή** ενός κόμβου απ' την αρχική ουρά.

ΔΙΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ

ΚΕΦΑΛΗ	100
--------	-----

ΟΥΡΑ	400
------	-----

prev	data	next
NULL	ΝΙΚΟΣ	200
100		

prev	data	next
100	ΚΩΣΤΑΣ	300
200		

prev	data	next
200	ΛΕΝΑ	400
300		

prev	data	next
300	ΒΙΚΥ	NULL
400		

β) Υποθέστε ότι η δομή αυτή υλοποιεί μια **διπλά συνδεδεμένη λίστα**. Διαθέτει δύο ειδικές μεταβλητές-δείκτες, την **ΚΕΦΑΛΗ** και την **ΟΥΡΑ** που προσφέρουν τη δυνατότητα προσπέλασης των κόμβων και από τις δύο κατευθύνσεις.

I) Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **εισαγωγή** μεταξύ των κόμβων με τιμές ΚΩΣΤΑΣ και ΛΕΝΑ ενός νέου κόμβου με τιμή ΤΑΣΟΣ που βρίσκεται αποθηκευμένη στη θέση μνήμης 450.

II) Να αναπαραστήσετε τη λίστα στη μνήμη μετά την **διαγραφή** του κόμβου με τιμή ΚΩΣΤΑΣ απ' την αρχική λίστα.

8. Δίνεται παρακάτω η απεικόνιση μιας δυναμικής δομής δεδομένων στη μνήμη ενός υπολογιστή. Η δομή αυτή υλοποιεί ένα γράφο (που διατηρεί τις ιδιότητες του **δυναμικού δένδρου αναζήτησης** δηλαδή κάθε κόμβος να συνδέεται με δύο κόμβους παιδιά του αλλά επιπλέον και με τον γονέα του (previous, εδώ : prev).

α) Να σχεδιάσετε την δενδρική μορφή του γράφου.

β) Να γράψετε τις διαδοχικές τιμές (data) των κόμβων (επίσκεψη κόμβων και διάσχιση του γράφου) που πρέπει να ελεγχθούν (για την ακρίβεια να συγκριθούν) έτσι ώστε να βρεθεί ο κόμβος που έχει τιμή (data) πλησιέστερη στην τιμή 96.

ΓΡΑΦΟΣ ΜΕ 3 ΔΕΙΚΤΕΣ ΣΕ ΚΑΘΕ ΚΟΜΒΟ

ΚΕΦΑΛΗ	1700
--------	------

prev	data	left	right
1100	10	NULL	NULL
1000			

prev	data	left	right
1300	20	1000	1200
1100			

prev	data	left	right
1100	30	NULL	NULL
1200			

prev	data	left	right
1700	40	1100	1500
1300			

prev	data	left	right
1500	50	NULL	NULL
1400			

prev	data	left	right
1300	60	1400	1600
1500			

prev	data	left	right
1500	70	NULL	NULL
1600			

prev	data	left	right
NULL	80	1300	2100
1700			

prev	data	left	right
1900	90	NULL	NULL
1800			

prev	data	left	right
2100	100	1800	2000
1900			

prev	data	left	right
1900	110	NULL	NULL
2000			

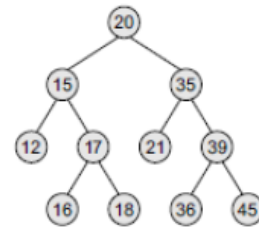
prev	data	left	right
1700	120	1900	2300
2100			

prev	data	left	right
2300	130	NULL	NULL
2200			

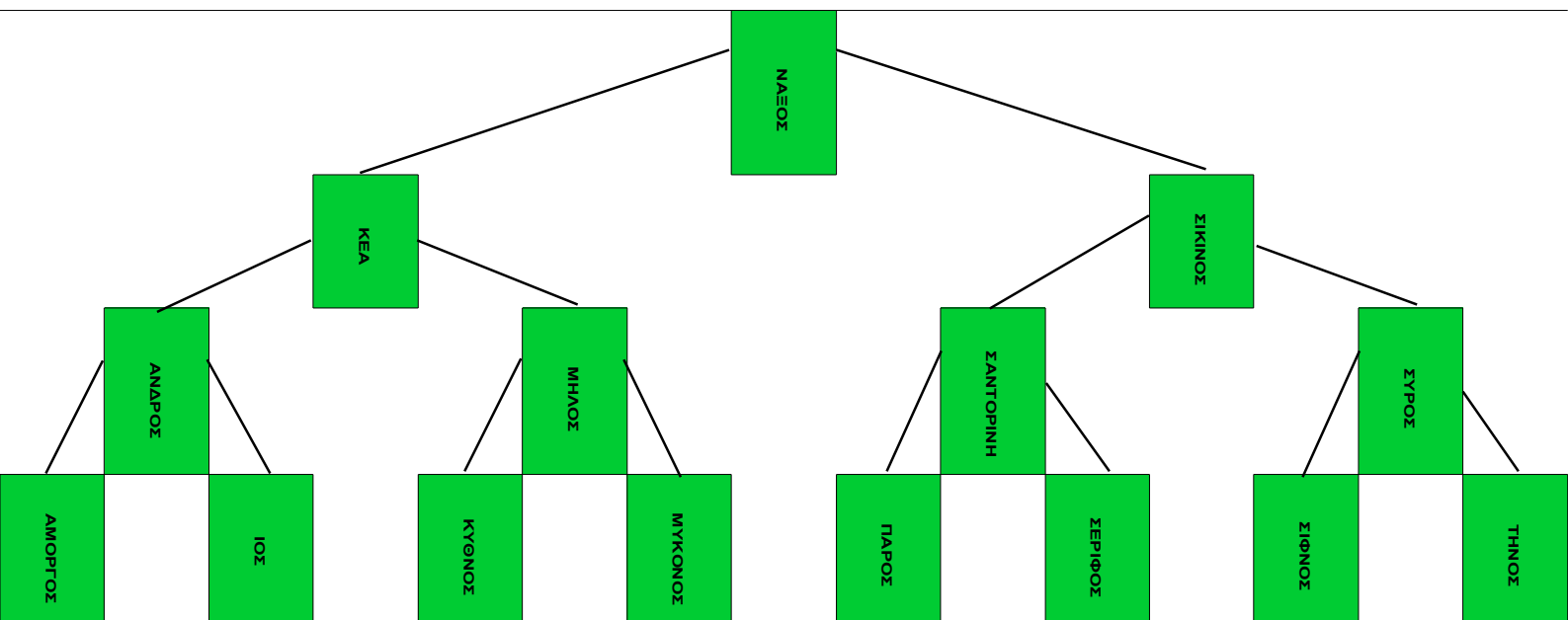
prev	data	left	right
2100	140	2200	2400
2300			

prev	data	left	right
2300	150	NULL	NULL
2400			

9. Ένα δυαδικό δένδρο μπορεί να αναπαρασταθεί στατικά με ένα μονοδιάστατο πίνακα ως εξής: Η ρίζα αποθηκεύεται στην πρώτη θέση του πίνακα. Για κάθε κόμβο (και για τη ρίζα) που είναι αποθηκευμένος στη θέση k του πίνακα, το αριστερό του παιδί αποθηκεύεται στη θέση $2k$ του πίνακα και το δεξί στη θέση $2k+1$. Σημειώνεται ότι για την αναπαράσταση ενός πλήρους δυαδικού δένδρου με n κόμβους σε πίνακα, απαιτούνται $2n+1$ θέσεις στον πίνακα.

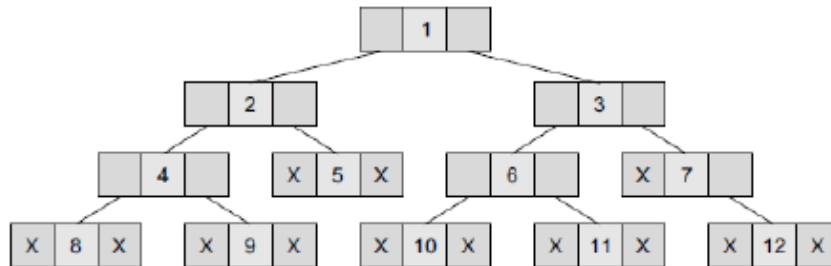


1	20
2	15
3	35
4	12
5	17
6	21
7	39
8	
9	
10	16
11	18
12	
13	
14	36
15	45



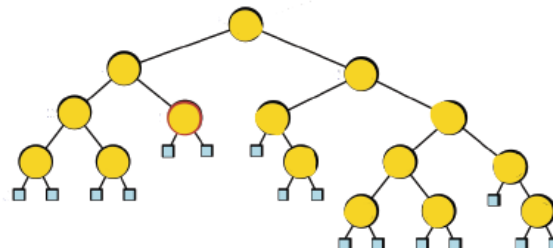
Να σχεδιάσετε μονοδιάστατο πίνακα 31 θέσεων ($31=2*15+1$) με τα ονόματα των 15 νησιών των Κυκλάδων, τα οποία αποτελούν τους κόμβους του δυαδικού δένδρου αναζήτησης που φαίνεται επάνω, τοποθετώντας τα στις θέσεις του πίνακα με τον τρόπο που περιγράφηκε παραπάνω. Στους κόμβους-φύλλα του δένδρου, επειδή δεν έχουν παιδιά, ως θέσεις των παιδιών τους στον πίνακα δώστε την τιμή 'NULL'.

10. Ένας άλλος τρόπος να αναπαριστούμε ένα δυαδικό δένδρο με πίνακα είναι να δημιουργούμε διδιάστατο τρίστηλο πίνακα και σε κάθε γραμμή του να καταχωρούμε την τιμή κάθε κόμβου του δένδρου και τις τιμές των θέσεων του πίνακα που είναι αποθηκευμένα το αριστερό και το δεξί παιδί του. Ένα παράδειγμα φαίνεται στο σχήμα που ακολουθεί. Στους κόμβους-φύλλα του δένδρου, επειδή δεν έχουν παιδιά, δίνεται η τιμή -1 ως θέσεις των παιδιών τους στον πίνακα. Να σχεδιάσετε αντίστοιχο διδιάστατο πίνακα για το δυαδικό δένδρο με τα νησιά της άσκησης 9. (Θεωρήστε κατ' εξαίρεση απ' ότι συμβαίνει στη 'Γλώσσα', ότι ο πίνακας μπορεί να δέχεται ακέραιες τιμές αλλά και τιμές τύπου χαρακτήρα).



	LEFT	DATA	RIGHT
ROOT 1	-1	8	-1
2	-1	10	-1
3	5	1	8
4			
5	9	2	14
6			
7			
8	20	3	11
9	1	4	12
10			
11	-1	7	18
12	-1	9	-1
13			
14	-1	5	-1
15			
16	-1	11	-1
17			
18	-1	12	-1
19			
20	2	6	16

11. Να τοποθετήσετε στο ακόλουθο δυαδικό δένδρο τις τιμές: 44, 5, 32, 62, 78, 1, 50, 88, 17, 3, 10, 48, 54, 4, 2 έτσι ώστε να αποτελεί δένδρο αναζήτησης.

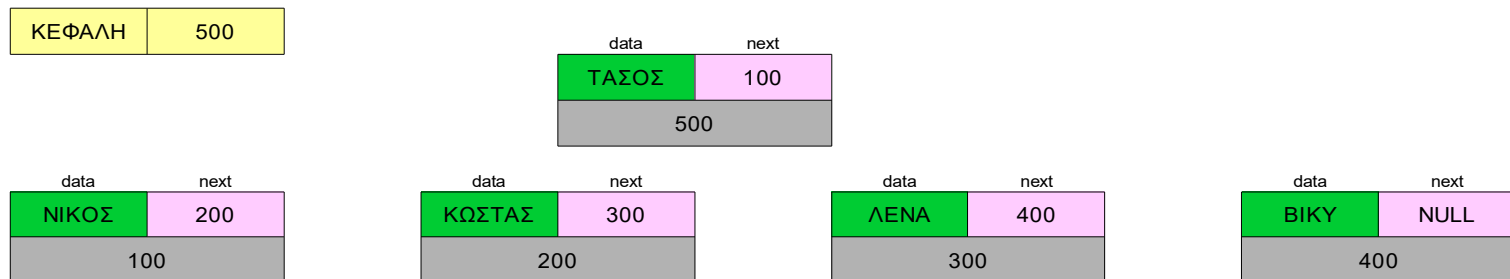


Επιμέλεια: Νικηφόρος Μανδηλαράς - ΓΕΛ Νάξου

Απαντήσεις

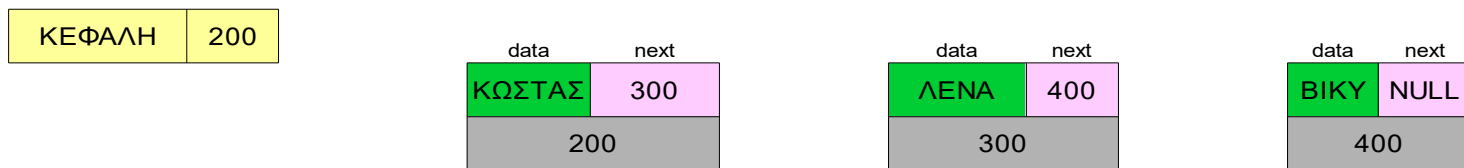
1. α) I Η λίστα υλοποιεί δυναμική στοίβα, επομένως η επεξεργασία της γίνεται με τη μέθοδο (LIFO). Ο δείκτης ΚΕΦΑΛΗ αποτελεί την κορυφή (top) της στοίβας. Ο κόμβος με τιμή ΝΙΚΟΣ είναι ο τελευταίος που ωθήθηκε στη στοίβα. Επομένως ο νέος κόμβος με τιμή ΤΑΣΟΣ ωθείται στην κορυφή της στοίβας. Έτσι η νέα αναπαράστασή της είναι:

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ



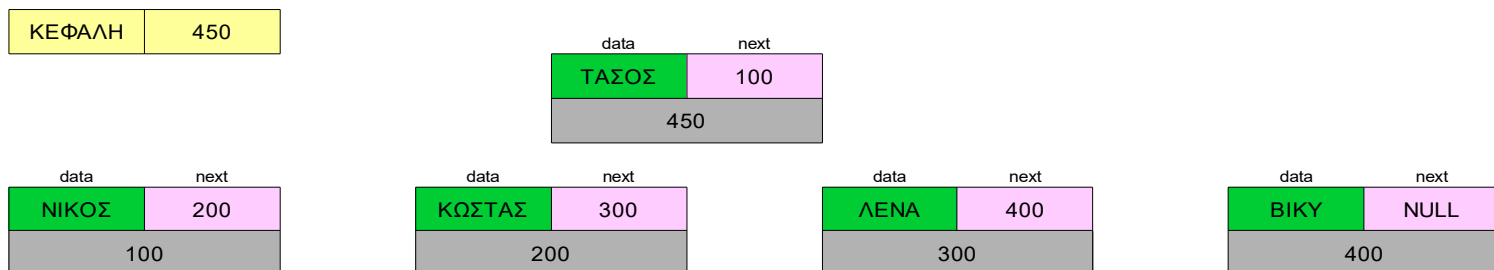
α) II Η απώθηση θα γίνει σύμφωνα με τη μέθοδο (LIFO). Ο δείκτης ΚΕΦΑΛΗ που αποτελεί την κορυφή (top) της στοίβας δείχνει τον κόμβο με τιμή ΝΙΚΟΣ. Δηλαδή ο κόμβος αυτός είναι ο τελευταίος που ωθήθηκε στη στοίβα. Επομένως ο κόμβος αυτός απωθείται από τη στοίβα. Ο δείκτης κεφαλή θα δείχνει τη θέση του επόμενου κόμβου με τιμή ΚΩΣΤΑΣ και η θέση που δέσμευε ο κόμβος ΝΙΚΟΣ ελευθερώνεται. (Για την ακρίβεια χάνεται απλώς η σύνδεση της ΚΕΦΑΛΗΣ με τον απωθημένο κόμβο. Σε πραγματικό προγραμματιστικό περιβάλλον, η θέση στη μνήμη, εφ' όσον το επιθυμούμε, απελευθερώνεται με ξεχωριστή εντολή). Έτσι η νέα αναπαράστασή της είναι:

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ



β) I Η λίστα υλοποιεί δυναμική ουρά, επομένως η επεξεργασία της γίνεται με τη μέθοδο (FIFO). Ο δείκτης ΚΕΦΑΛΗ αποτελεί τον δείκτη **rear** της ουράς. Ο κόμβος με τιμή ΝΙΚΟΣ είναι ο τελευταίος που εισάχθηκε στην ουρά ενώ ο κόμβος με τιμή ΒΙΚΥ εισάχθηκε πρώτος. Επομένως ο νέος κόμβος με τιμή ΤΑΣΟΣ εισάγεται στο τέλος της ουράς. Έτσι η νέα αναπαράστασή της είναι:

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ



β) II Η εξαγωγή θα γίνει σύμφωνα με τη μέθοδο (FIFO). Ο κόμβος με τιμή ΒΙΚΥ είναι αυτός που εισάχθηκε πρώτος και συνεπώς αυτός πρέπει να εξαχθεί. Η διεύθυνση μνήμης του αποτελεί ουσιαστικά την τιμή του δείκτη **front**. Η προσπέλαση όμως σ' αυτόν δεν μπορεί να γίνει απευθείας, αλλά ξεκινώντας απ' τον κόμβο που δείχνει η ΚΕΦΑΛΗ θα πρέπει να διασχίσουμε όλη τη λίστα μέχρι τον κόμβο του οποίου ο δείκτης next να έχει τιμή NULL. Αυτό οφείλεται στο γεγονός ότι έχουμε υλοποίηση της δυναμικής ουράς με απλά συνδεδεμένη λίστα και δεν έχουμε δείκτη που να δείχνει άμεσα τον πρώτο κόμβο (front).

(Βλέπε άσκηση 7α όπου γίνεται υλοποίηση δυναμικής ουράς με Διπλά συνδεδεμένη Λίστα). Ο δείκτης next του κόμβου με τιμή ΛΕΝΑ θα πάρει την τιμή NULL και η θέση μνήμης που δέσμευε ο κόμβος ΒΙΚΥ ελευθερώνεται. (Για την ακρίβεια χάνεται απλώς η σύνδεση της ουράς με τον εξαχθέντα κόμβο. Σε πραγματικό προγραμματιστικό περιβάλλον, η θέση στη μνήμη, εφ' όσον το επιθυμούμε, απελευθερώνεται με ξεχωριστή εντολή). Έτσι η αναπαράστασή της μετά την εξαγωγή είναι:

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ

ΚΕΦΑΛΗ	100
--------	-----

data	next
ΝΙΚΟΣ	200
100	

data	next
ΚΩΣΤΑΣ	300
200	

data	next
ΛΕΝΑ	NULL
300	

γ) I Εφ' όσον ο νέος κόμβος με τιμή ΤΑΣΟΣ πρέπει να εισαχθεί μετά τον κόμβο με τιμή ΛΕΝΑ, θα πρέπει ο δείκτης του κόμβου ΛΕΝΑ να δείχνει στον νέο κόμβο ΤΑΣΟΣ και αυτός να δείχνει στον κόμβο με τιμή ΒΙΚΥ. Έτσι η νέα αναπαράστασή της είναι:

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ

ΚΕΦΑΛΗ	100
--------	-----

data	next
ΤΑΣΟΣ	400
450	

data	next
ΝΙΚΟΣ	200
100	

data	next
ΚΩΣΤΑΣ	300
200	

data	next
ΛΕΝΑ	450
300	

data	next
ΒΙΚΥ	NULL
400	

γ) II Εφ' όσον ο κόμβος με τιμή ΛΕΝΑ πρέπει να διαγραφεί από τη λίστα, θα πρέπει ο δείκτης του προηγούμενου κόμβου ΚΩΣΤΑΣ να δείχνει απευθείας στον κόμβο ΒΙΚΥ και η μνήμη που καταλαμβάνει ο κόμβος ΛΕΝΑ να αποδεσμευτεί. (Για την ακρίβεια χάνεται απλώς η σύνδεση της λίστας με τον διαγραφέντα κόμβο. Σε πραγματικό προγραμματιστικό περιβάλλον, η θέση στη μνήμη, εφ' όσον το επιθυμούμε, απελευθερώνεται με ξεχωριστή εντολή). Έτσι η νέα αναπαράσταση της μνήμης είναι:

ΑΠΛΑ ΣΥΝΔΕΔΕΜΕΝΗ ΛΙΣΤΑ

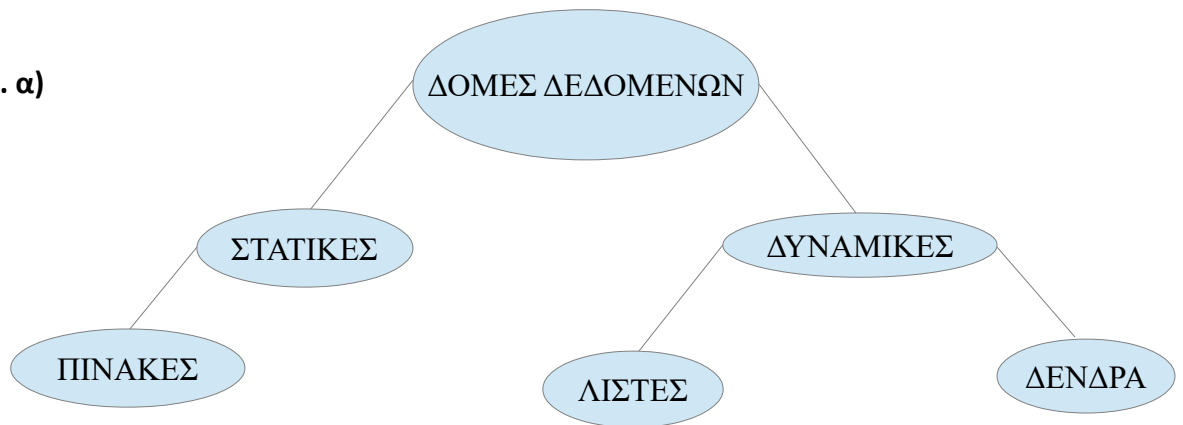
ΚΕΦΑΛΗ	100
--------	-----

data	next
ΝΙΚΟΣ	200
100	

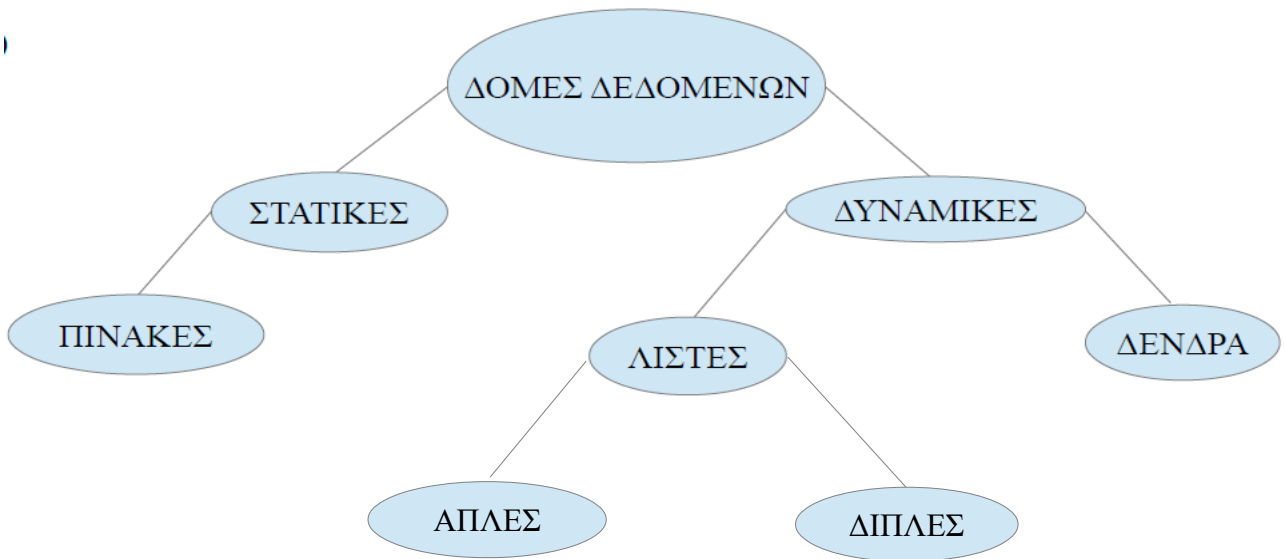
data	next
ΚΩΣΤΑΣ	400
200	

data	next
ΒΙΚΥ	NULL
400	

2. α)



β) Ι



β) ΙΙ

ΚΕΦΑΛΗ	10
--------	----

left	data	right
20	Δομές Δεδομένων	30
10		

left	data	right
40	Στατικές	NULL
20		

left	data	right
50	Δυναμικές	60
30		

left	data	right
NULL	Πίνακες	NULL
40		

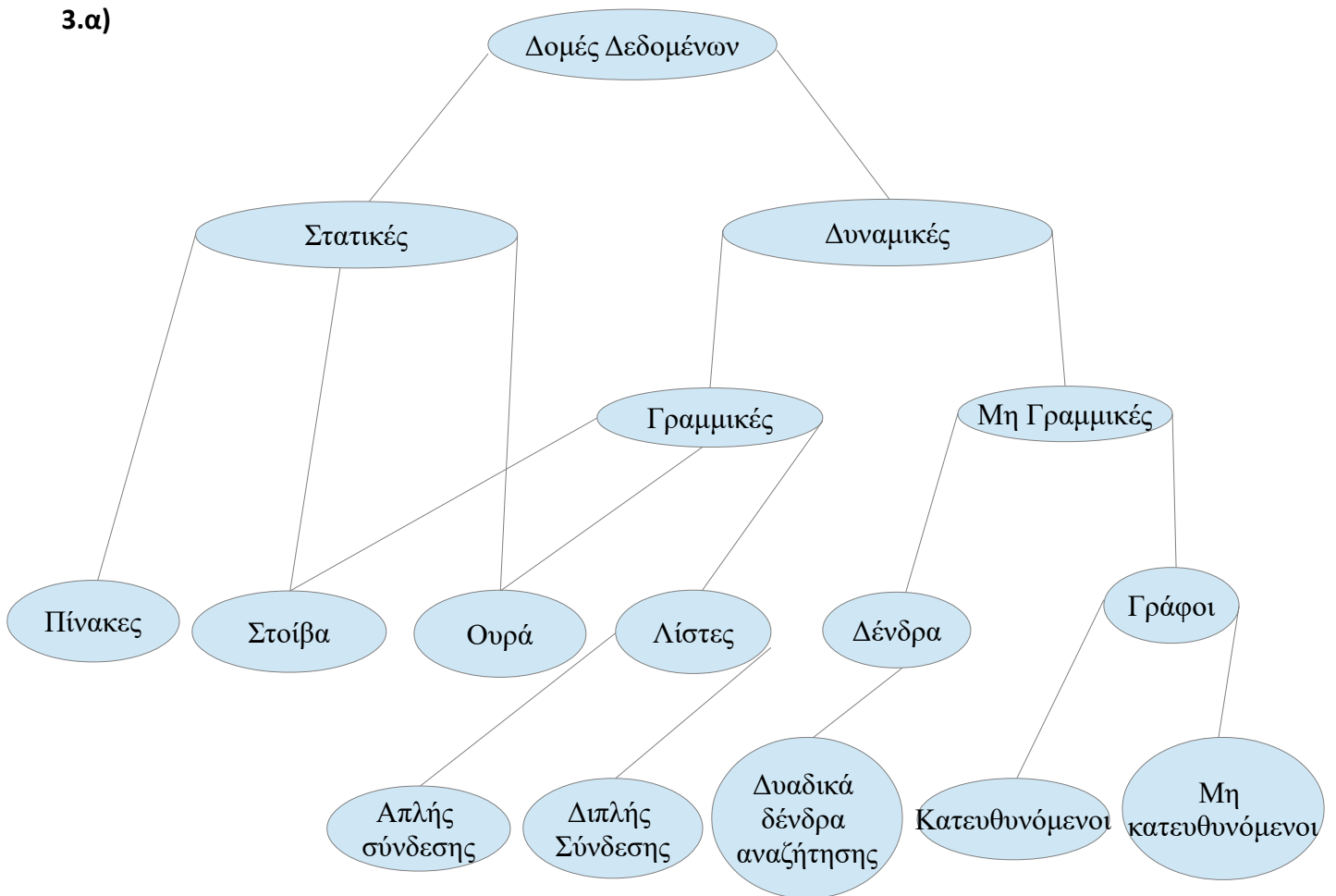
left	data	right
70	Λίστες	80
50		

left	data	right
NULL	Δένδρα	NULL
60		

left	data	right
NULL	Απλές	NULL
70		

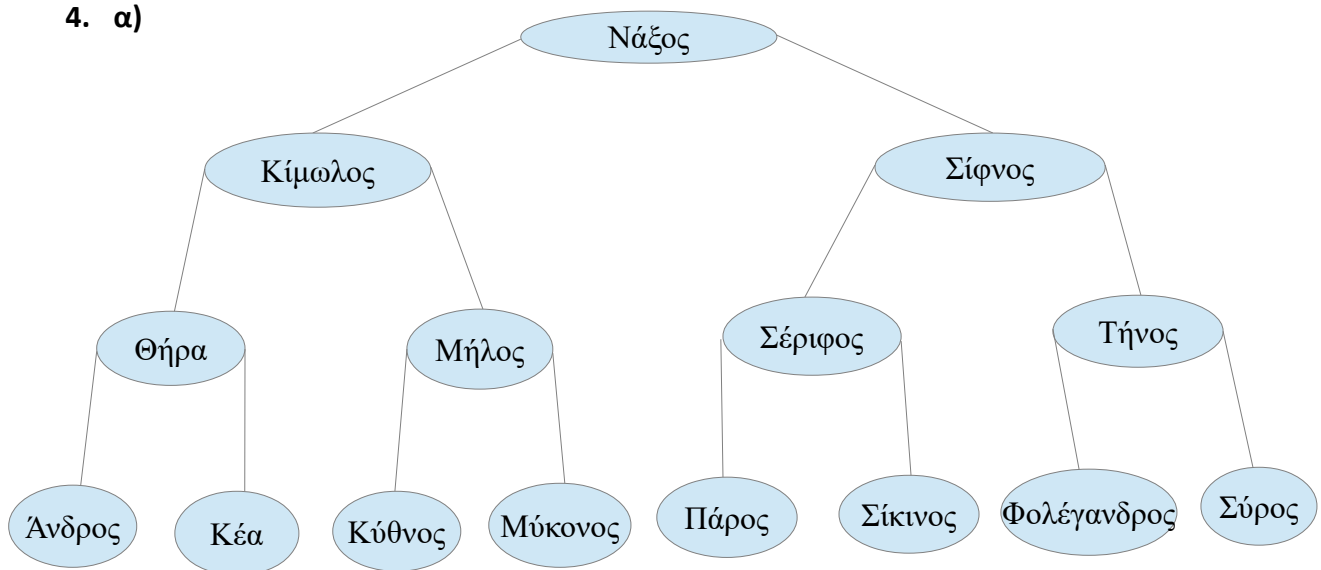
left	data	right
NULL	Διπλές	NULL
80		

3.α)



β) Η στοιβά και η ουρά μπορούν να υλοποιηθούν με χρήση πινάκων δηλαδή αποτελούν στατικές, δομές δεδομένων, αλλά μπορούν να υλοποιηθούν και με χρήση συνδεδεμένων λιστών δηλαδή μπορούν να χαρακτηριστούν και δυναμικές δομές δεδομένων. Εφ' όσον λοιπόν στο παραπάνω δενδρικό διάγραμμα οι κόμβοι Στοιβά και Ουρά έχουν δύο γονείς, η δομή που απεικονίζεται αποτελεί γράφο.

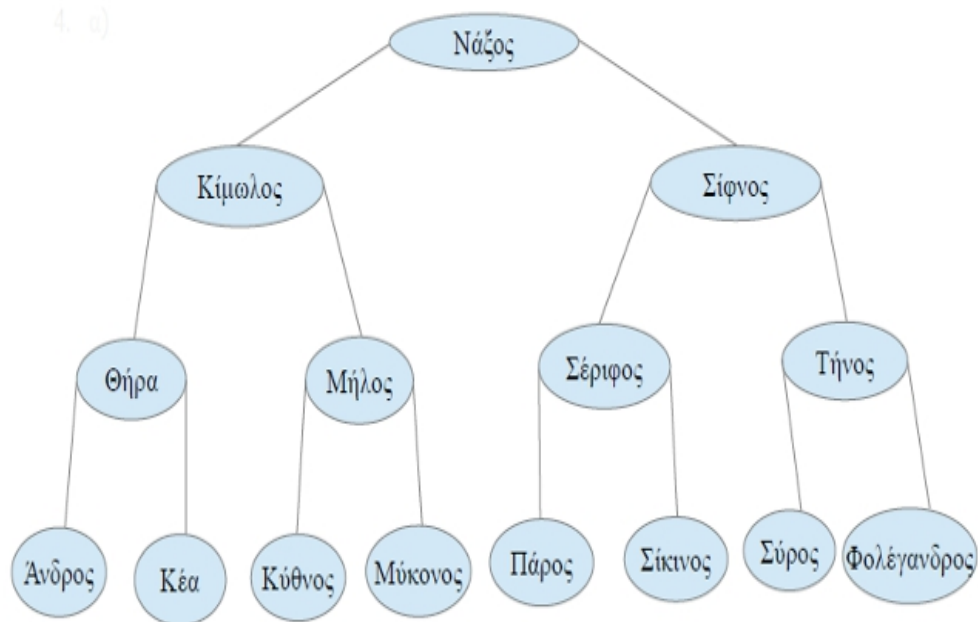
4. α)



Επιμέλεια: Νικηφόρος Μανδηλαράς - ΓΕΛ Νάξου

β) Για να είναι δυαδικό δένδρο αναζήτησης πρέπει **κάθε κόμβος του να έχει μεγαλύτερη τιμή από κάθε αριστερό υποδένδρο του και μικρότερη από κάθε δεξί υποδένδρο του**. Οι κόμβοι που δεν ικανοποιούν τη συνθήκη είναι: ο Φολέγανδρος ($>$ από Τήνος ενώ ανήκει στο αριστερό υποδένδρο του) και ο Σύρος ($<$ από Τήνος ενώ ανήκει στο δεξί υποδένδρο του). Άρα πρέπει να αλλάξουν αμοιβαία τη θέση τους οι κόμβοι Φολέγανδρος και Σύρος. Αντίστοιχα στη μνήμη στον κόμβο Τήνος πρέπει να αντιμεταθέσουμε τις τιμές των δύο δεικτών του.

Έτσι το δένδρο γίνεται :



και η απεικόνιση της δομής που το υλοποιεί στη μνήμη:

ΔΕΝΔΡΙΚΗ ΔΟΜΗ ΜΕ 2 ΔΕΙΚΤΕΣ ΣΕ ΚΑΘΕ ΚΟΜΒΟ

ΚΕΦΑΛΗ	10
--------	----

data	left	right
ΝΑΞΟΣ	20	30
10		

data	left	right
ΚΙΜΩΛΟΣ	40	50
20		

data	left	right
ΣΙΦΝΟΣ	60	70
30		

data	left	right
ΘΗΡΑ	80	90
40		

data	left	right
ΜΗΛΟΣ	100	110
50		

data	left	right
ΣΕΡΙΦΟΣ	120	130
60		

data	left	right
ΤΗΝΟΣ	140	150
70		

data	left	right
ΑΝΔΡΟΣ	NULL	NULL
80		

data	left	right
ΚΕΑ	NULL	NULL
90		

data	left	right
ΚΥΘΝΟΣ	NULL	NULL
100		

data	left	right
ΜΥΚΟΝΟΣ	NULL	NULL
110		

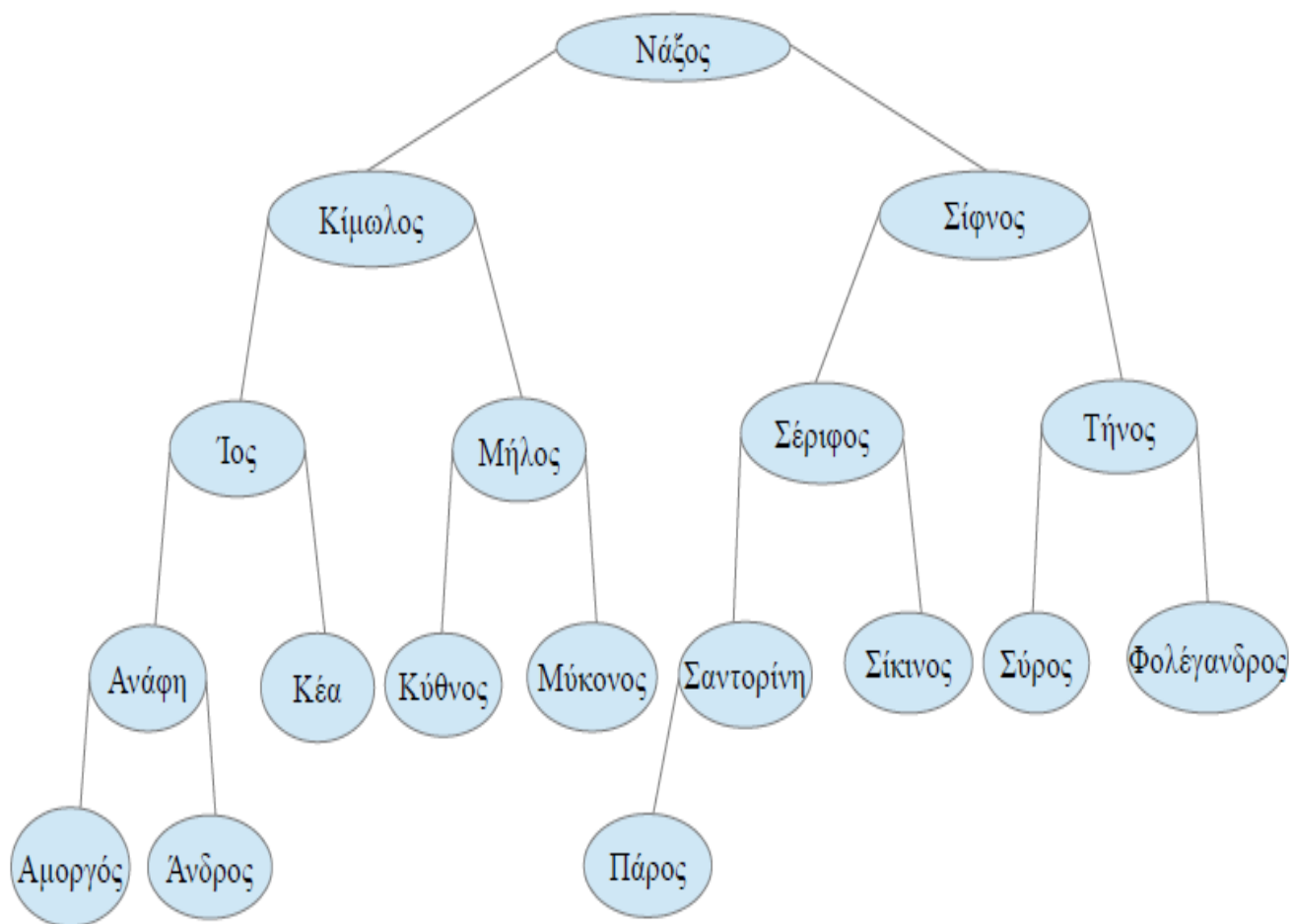
data	left	right
ΠΑΡΟΣ	NULL	NULL
120		

data	left	right
ΣΙΚΙΝΟΣ	NULL	NULL
130		

data	left	right
ΣΥΡΟΣ	NULL	NULL
140		

data	left	right
ΦΟΛΕΓΑΝΔΡΟΣ	NULL	NULL
150		

γ) το νέο δένδρο γίνεται :



δ) και η νέα απεικόνιση στη μνήμη, της δομής που το υλοποιεί:

ΔΕΝΔΡΙΚΗ ΔΟΜΗ ΜΕ 2 ΔΕΙΚΤΕΣ ΣΕ ΚΑΘΕ ΚΟΜΒΟ

ΚΕΦΑΛΗ	10
--------	----

data	left	right
ΝΑΞΟΣ	20	30
10		

data	left	right
ΚΙΜΩΛΟΣ	180	50
20		

data	left	right
ΣΙΦΝΟΣ	60	70
30		

data	left	right
ΣΑΝΤΟΡΙΝΗ	120	NULL
40		

data	left	right
ΜΗΛΟΣ	100	110
50		

data	left	right
ΣΕΡΙΦΟΣ	40	130
60		

data	left	right
ΤΗΝΟΣ	140	150
70		

data	left	right
ΑΝΔΡΟΣ	NULL	NULL
80		

data	left	right
ΚΕΑ	NULL	NULL
90		

data	left	right
ΚΥΘΝΟΣ	NULL	NULL
100		

data	left	right
ΜΥΚΟΝΟΣ	NULL	NULL
110		

data	left	right
ΠΑΡΟΣ	NULL	NULL
120		

data	left	right
ΣΙΚΙΝΟΣ	NULL	NULL
130		

data	left	right
ΣΥΡΟΣ	NULL	NULL
140		

data	left	right
ΦΟΛΕΓΑΝΔΡΟΣ	NULL	NULL
150		

data	left	right
ΑΜΟΡΓΟΣ	NULL	NULL
160		

data	left	right
ΑΝΑΦΗ	160	80
170		

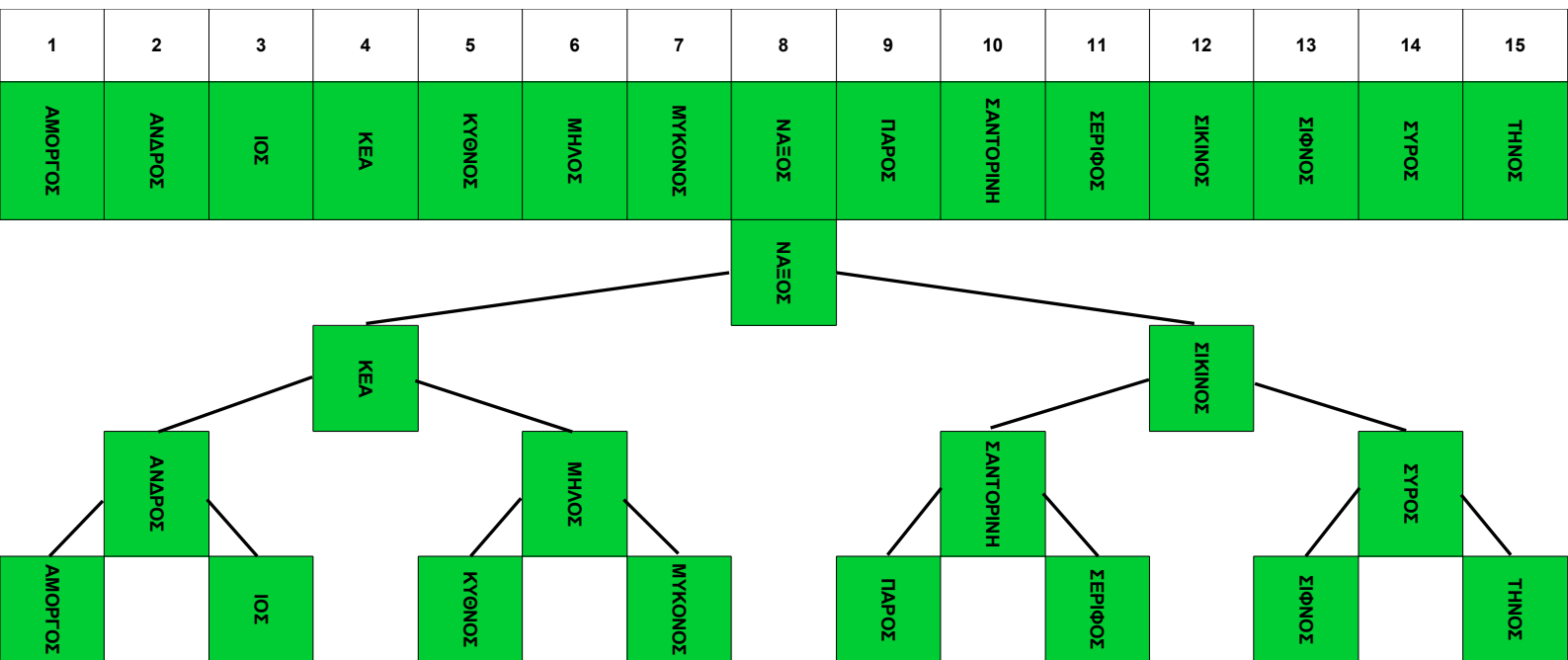
data	left	right
ΙΟΣ	170	90
180		

Επιμέλεια: Νικηφόρος Μανδηλαράς - ΓΕΛ Νάξου

ε) Απαιτούνται πέντε συγκρίσεις :

1. η διάταξη (Πάρος > Νάξος) μας οδηγεί στο δεξί υποδένδρο
2. η διάταξη (Πάρος < Σίφνος) μας οδηγεί στο αριστερό υποδένδρο
3. η διάταξη (Πάρος < Σέριφος) μας οδηγεί στο αριστερό υποδένδρο
4. η διάταξη (Πάρος < Σαντορίνη) μας οδηγεί στο αριστερό υποδένδρο
5. εύρεση του ζητούμενου κόμβου (Πάρος = Πάρος)
(αποτελεί τη χειρότερη περίπτωση αναζήτησης όπως και η αναζήτηση των φύλλων Άνδρος και Αμοργός).

5. α)



Παρατηρήστε ότι το μεσαίο στοιχείο κάθε υποπίνακα που προκύπτει μετά από τη δυαδική διαίρεση του γονικού του υποπίνακα, ξεκινώντας απ' τον αρχικό πίνακα, αντιστοιχεί σε ρίζα του αντίστοιχου υποδένδρου του δυαδικού δένδρου αναζήτησης.

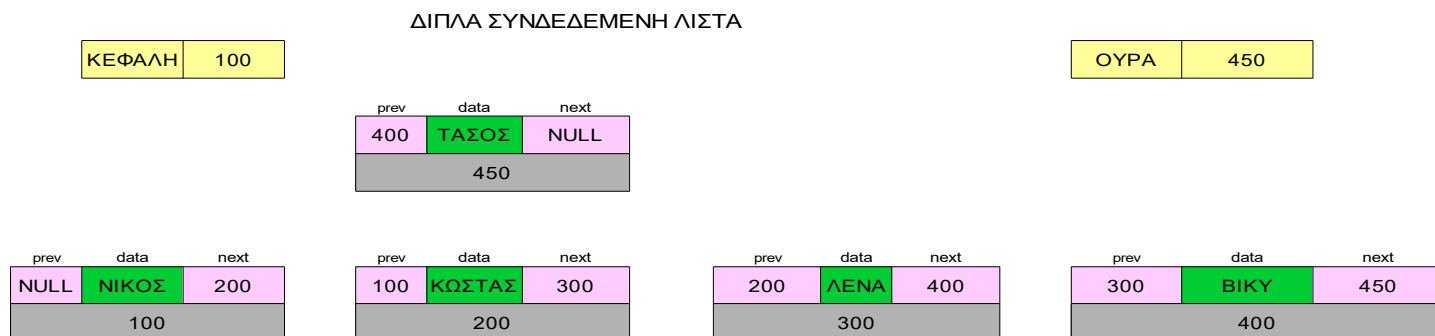
β) Στην χειρότερη περίπτωση απαιτούνται 4 έλεγχοι, όταν ο ζητούμενος κόμβος αποτελεί φύλλο αυτού του δένδρου.

Το ίδιο πλήθος ελέγχων απαιτούνται και στον ταξινομημένο πίνακα διότι η επίσκεψη κάθε κόμβου του δυαδικού δένδρου αναζήτησης, ισοδυναμεί με την προσπέλαση του μεσαίου στοιχείου, κάθε δυαδικά διαιρούμενου υποπίνακα, του αρχικού πίνακα (ο γνωστός αλγόριθμος δυαδικής αναζήτησης στοιχείου σε ταξινομημένο πίνακα-Διαίρει και Βασίλευε!).

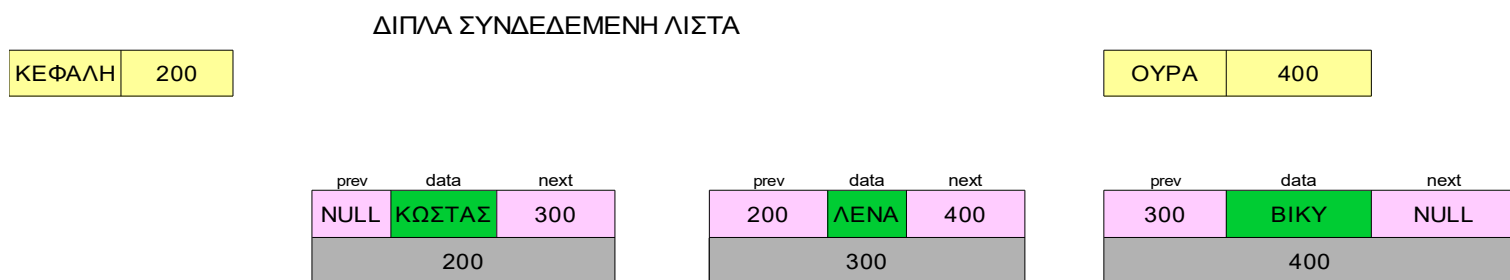
6. Οι θέσεις μνήμης των κόμβων θα πρέπει να εισαχθούν στον πίνακα όπως φαίνεται παρακάτω:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
80	40	90	20	100	50	110	10	120	60	130	30	140	70	150

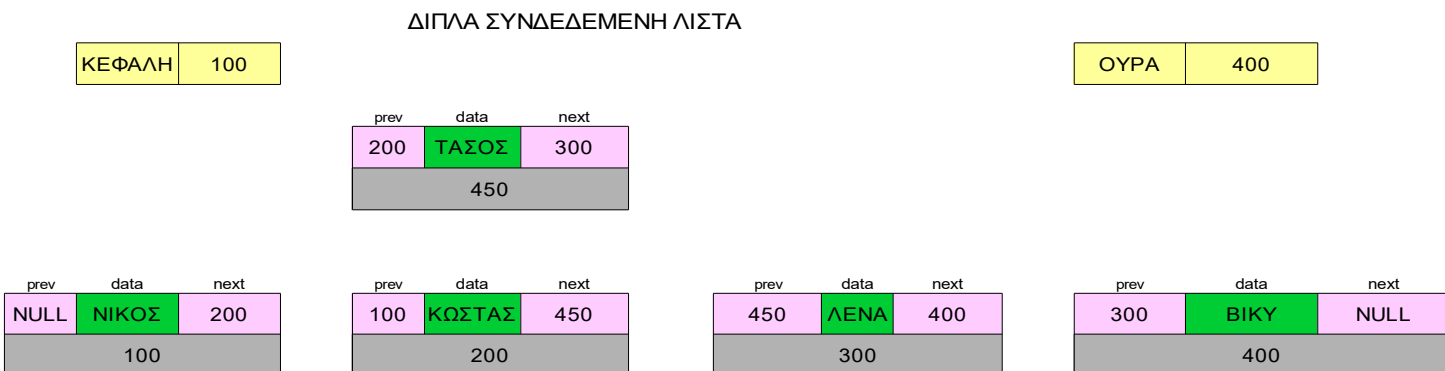
7. α I) Η εισαγωγή του νέου κόμβου ΤΑΣΟΣ στην ουρά γίνεται στο πίσω μέρος της, έτσι ο δείκτης **ΟΥΡΑ** θα δείχνει τον κόμβο ΤΑΣΟΣ που θα αποτελεί πλέον τον πίσω κόμβο. Η νέα αναπαράσταση της λίστας στη μνήμη θα είναι:



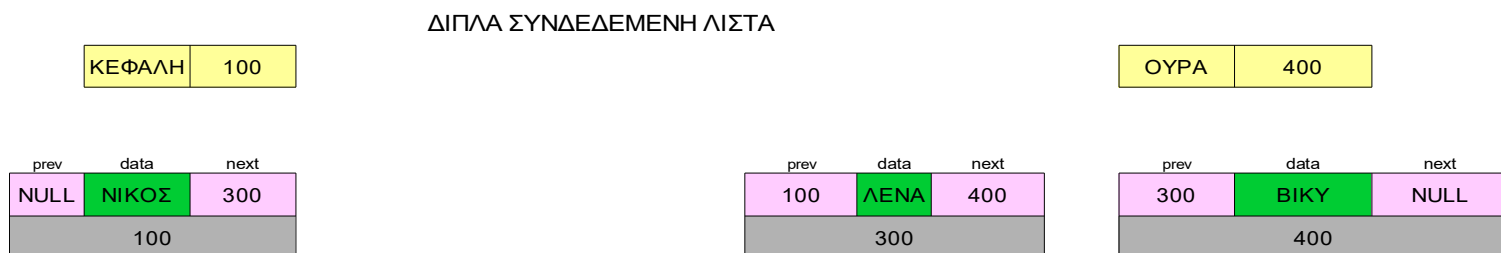
α II) Η εξαγωγή κόμβου από ουρά γίνεται απ' το εμπρός μέρος της, έτσι ο δείκτης ΚΕΦΑΛΗ θα δείχνει τον κόμβο ΚΩΣΤΑΣ που θα αποτελεί πλέον τον εμπρός κόμβο και η μνήμη που καταλάμβανε ο κόμβος ΝΙΚΟΣ αποδεσμεύεται. (Για την ακρίβεια χάνεται απλώς η σύνδεση της ΚΕΦΑΛΗΣ με τον εξαχθέντα κόμβο. Σε πραγματικό προγραμματιστικό περιβάλλον, η θέση στη μνήμη, εφ' όσον το επιθυμούμε, απελευθερώνεται με ξεχωριστή εντολή). Η νέα αναπαράσταση της λίστας στη μνήμη θα είναι:



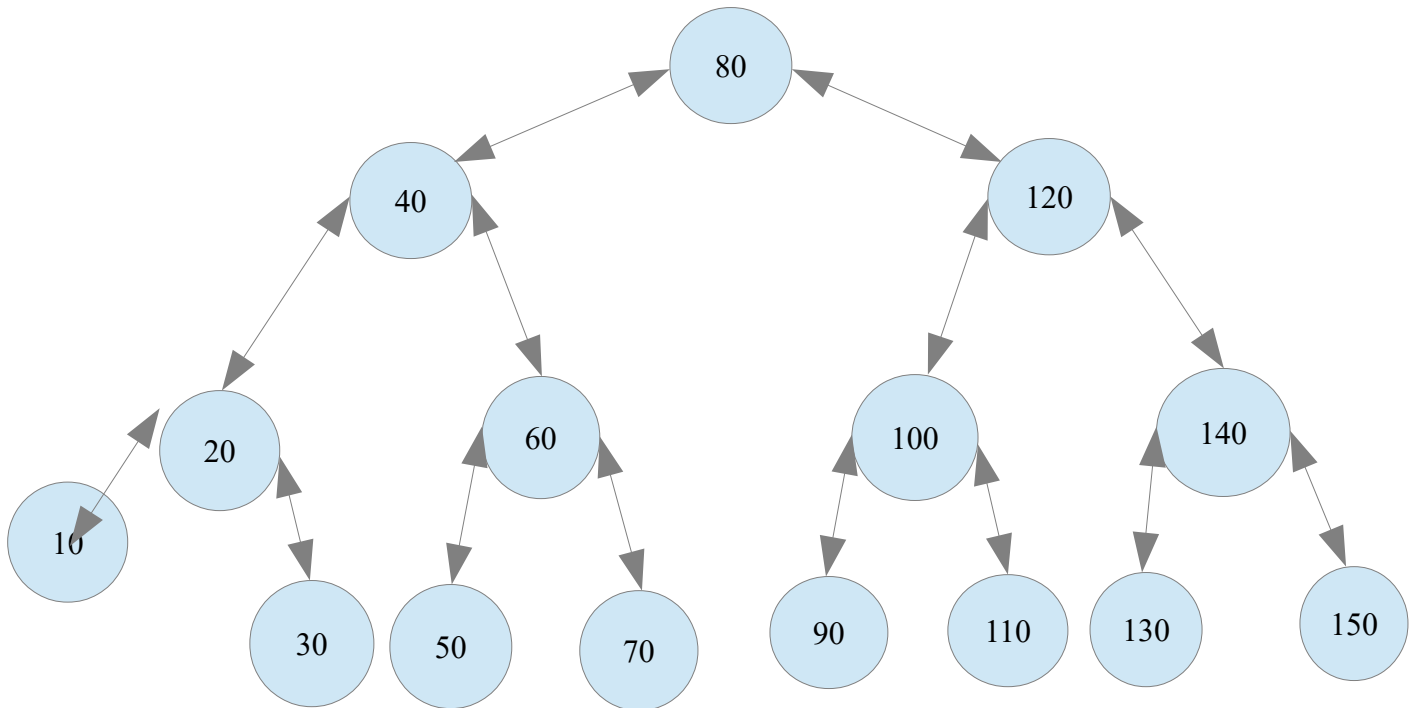
β I)



β II)



8. α) Ο κόμβος με τιμή 80 είναι ο κόμβος που δείχνει η κεφαλή, οπότε λειτουργεί σαν ρίζα του δένδρου αναζήτησης-γράφου (αρχικός κόμβος του γράφου).



β) Επίσκεψη με τη βοήθεια της ΚΕΦΑΛΗΣ στη ρίζα 80 (θέση μνήμης 1700),
η σύγκριση $96 > 80$ μας οδηγεί σε επίσκεψη στο δεξί υποδένδρο,
η σύγκριση $96 < 120$ μας οδηγεί σε επίσκεψη στο αριστερό υποδένδρο,
η σύγκριση $96 < 100$ μας οδηγεί σε επίσκεψη στο αριστερό υποδένδρο,
η σύγκριση $96 > 90$ μας οδηγεί στο δεξί υποδένδρο,
όμως δεξί παιδί δεν υπάρχει, η τιμή του δεξιού δείκτη είναι NULL (φύλλο του δένδρου).
Υπολογισμός διαφοράς : $[96 - 90 = 6]$
Επιστροφή και επίσκεψη στο γονέα του κόμβου 90, δηλαδή στον κόμβο 100.
Υπολογισμός διαφοράς : $[96 - 100 = -4]$.
Άρα την πλησιέστερη τιμή του δένδρου, στην τιμή 96, έχει ο κόμβος 100. ($|6| > |-4|$)

Τελικά οι κόμβοι επίσκεψης ήταν οι κόμβοι με τιμές : 80, 120, 100, 90, 100

Παρατήρηση:

Με την υλοποίηση του γράφου έτσι ώστε κάθε κόμβος να συνδέεται με δύο κόμβους παιδιά του αλλά και με τον γονέα του (previous, εδώ : prev) έχουμε τη δυνατότητα να διασχίσουμε το γράφο και αντίστροφα. Έτσι μπορούμε στην άσκηση να επιστρέψουμε απ' τον κόμβο 90 στον κόμβο 100 (δηλαδή από παιδί σε γονέα), για να υπολογίσουμε ποια διαφορά είναι μικρότερη.

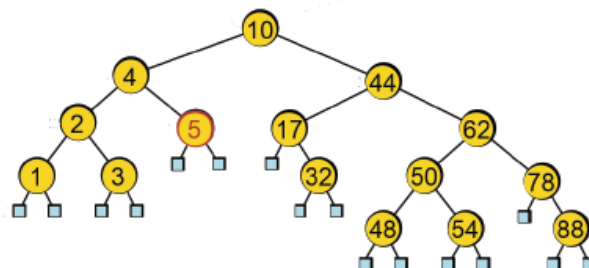
9.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ΝΑΞΟΣ	ΚΕΑ	ΣΙΚΙΝΟΣ	ΑΝΔΡΟΣ	ΜΗΛΟΣ	ΣΑΝΤΟΡΙΝΗ	ΣΥΡΟΣ	ΑΜΟΡΓΟΣ	ΙΟΣ	ΚΥΘΝΟΣ	ΜΥΚΟΝΟΣ	ΠΑΡΟΣ	ΣΕΡΙΦΟΣ	ΣΙΦΝΟΣ	ΤΗΝΟΣ	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

10.

	LEFT	DATA	RIGHT
1	2	ΝΑΞΟΣ	3
2	4	ΚΕΑ	5
3	6	ΣΙΚΙΝΟΣ	7
4	8	ΑΝΔΡΟΣ	9
5	10	ΜΗΛΟΣ	11
6	12	ΣΑΝΤΟΡΙΝΗ	13
7	14	ΣΥΡΟΣ	15
8	-1	ΑΜΟΡΓΟΣ	-1
9	-1	ΙΟΣ	-1
10	-1	ΚΥΘΝΟΣ	-1
11	-1	ΜΥΚΟΝΟΣ	-1
12	-1	ΠΑΡΟΣ	-1
13	-1	ΣΕΡΙΦΟΣ	-1
14	-1	ΣΙΦΝΟΣ	-1
15	-1	ΤΗΝΟΣ	-1

11.



Επιμέλεια: Νικηφόρος Μανδηλαράς - ΓΕΛ Νάξου