



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

Αλγόριθμοι

Διάλεξη 9: Δομές δεδομένων – Δένδρο

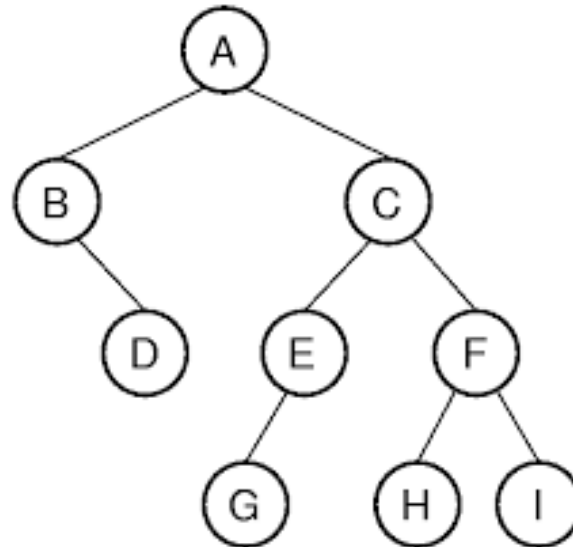
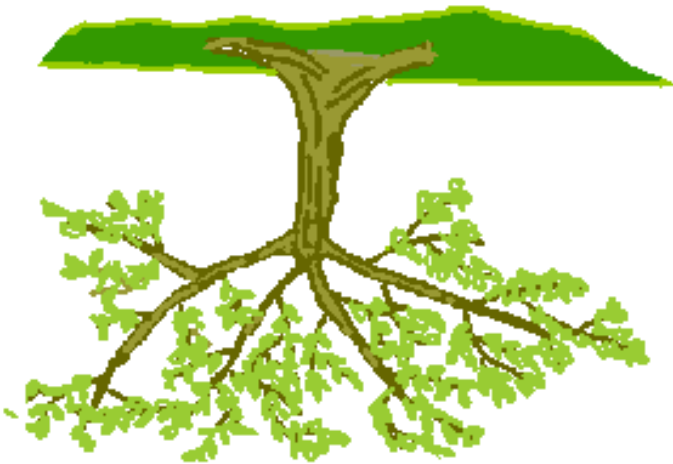
Τσούρος Δήμος

ΠΕΡΙΕΧΟΜΕΝΑ

- **Εισαγωγή**
- **Δένδρο**
- **Δυαδικό δένδρο αναζήτησης**
- **Αλγόριθμος δυαδικού δένδρου αναζήτησης**

ΔΕΝΔΡΟ (1)

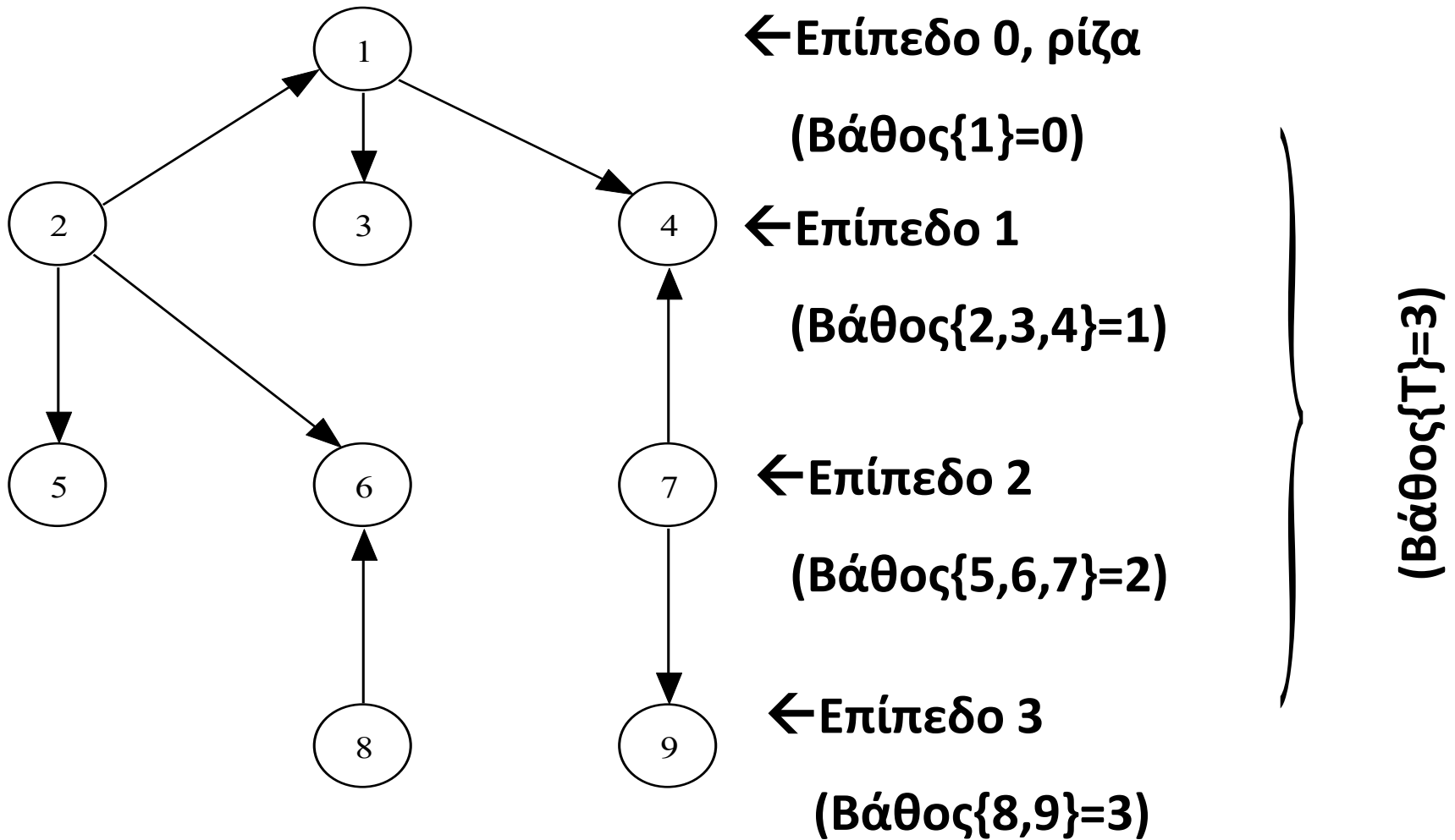
- **Δένδρο** (tree): είναι ένας συνεκτικό γράφημα (με κόμβους και ακμές) χωρίς κύκλους. Σημαντική δομή δεδομένων για πολλούς αλγορίθμους!
- Ιδιότητες
 - Κάθε δύο κόμβοι συνδέονται με ακριβώς ένα δρόμο
 - Ένα δένδρο με n κόμβους έχει $n - 1$ ακμές



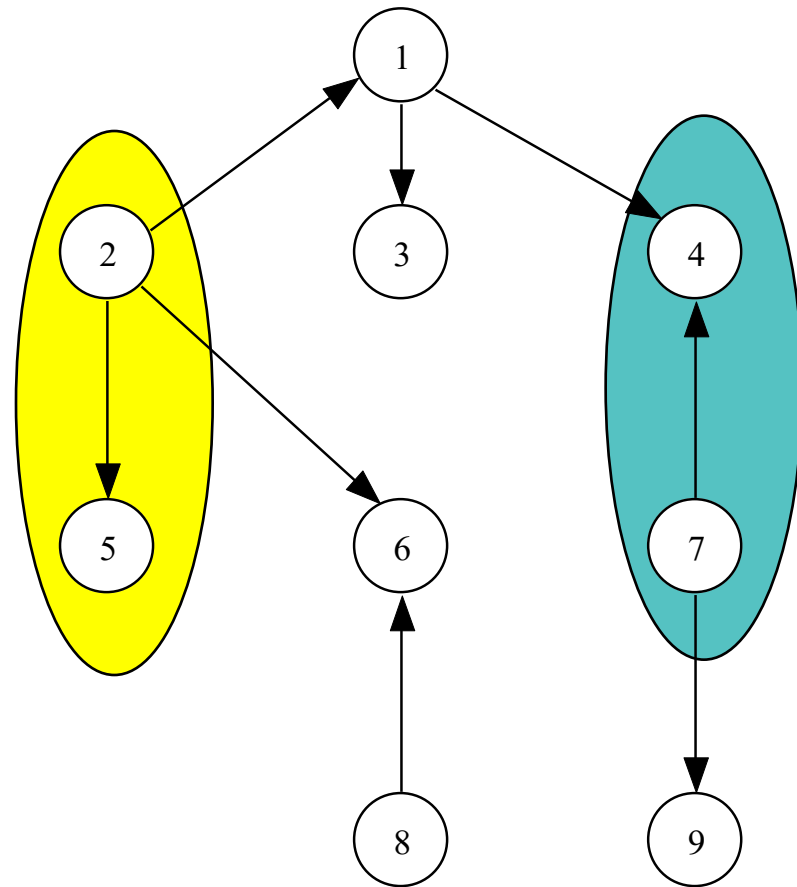
ΔΕΝΔΡΟ (2)

- **Διαδρομή** (path) από ένα κόμβο x σε ένα άλλο κόμβο y είναι μια ακολουθία κόμβων όπου πρώτος είναι ο x , τελευταίος ο y και κάθε άλλος κόμβος είναι παιδί του προηγούμενου
- **Μήκος** (length) μιας διαδρομής είναι ο αριθμός των ακμών που περιλαμβάνονται σ' αυτήν
- **Επίπεδο** (level) ή **βάθος** (depth) ενός κόμβου είναι το μήκος της μοναδικής διαδρομής από τη ρίζα στον κόμβο αυτό
- **Βαθμός** (degree) ενός κόμβου είναι ο αριθμός των παιδιών του
- **Βαθμός** ενός δένδρου είναι ο μέγιστος των βαθμών των κόμβων του

ΔΕΝΔΡΟ (3)



ΔΕΝΔΡΟ (5)

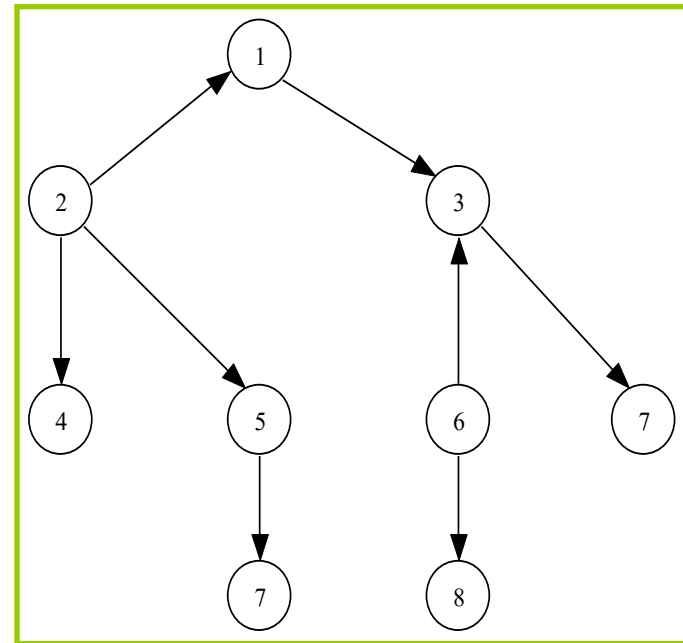
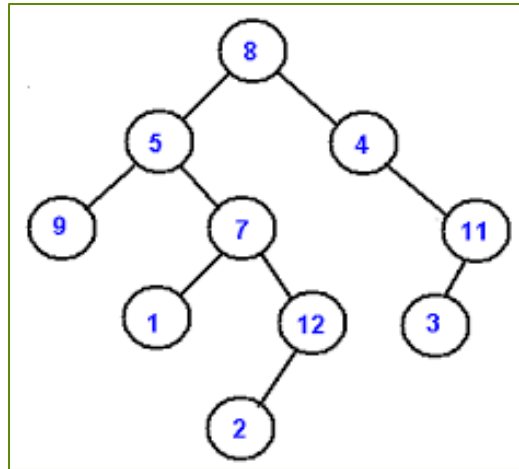
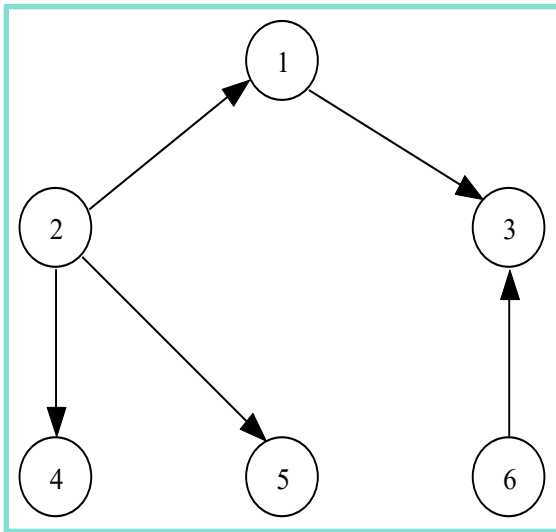


$(2, 5) \in T$,
Κόμβος 2 πατέρας του κόμβου 5
Κόμβος 5 παιδί του κόμβου 2

$(7, 4) \in T$,
Κόμβος 4 πατέρας του κόμβου 7
Κόμβος 7 παιδί του κόμβου 4

ΔΕΝΔΡΟ (6)

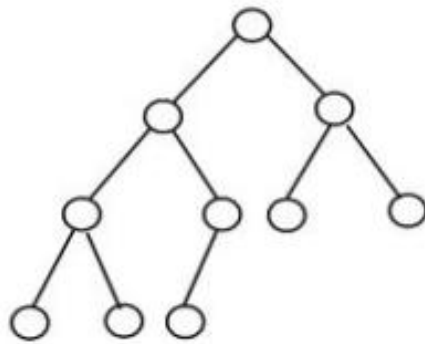
Αν κάθε κόμβος ενός δένδρου έχει το πολύ δυο παιδιά, το δέντρο ονομάζεται **δυναδικό** (*binary tree*)



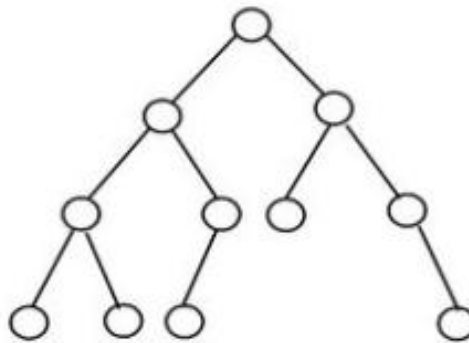
Δυναδικά δένδρα

ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (3)

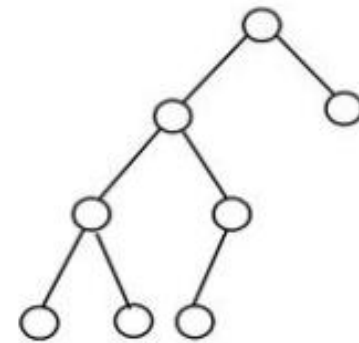
Αν κάθε επίπεδο βάθους t , $1 \leq t \leq h - 1$, ενός δένδρου T αποτελείται από το μέγιστο δυνατό πλήθος κόμβων 2^t το δένδρο T ονομάζεται **πλήρες** (*complete tree*), δηλαδή αν σε όλα τα επίπεδα (εκτός ίσως από το τελευταίο) έχει το μέγιστο δυνατό αριθμό κόμβων και όλοι οι κόμβοι στο τελευταίο επίπεδο βρίσκονται όσο το δυνατό στα αριστερά του δένδρου



(α)



(β)



(γ)

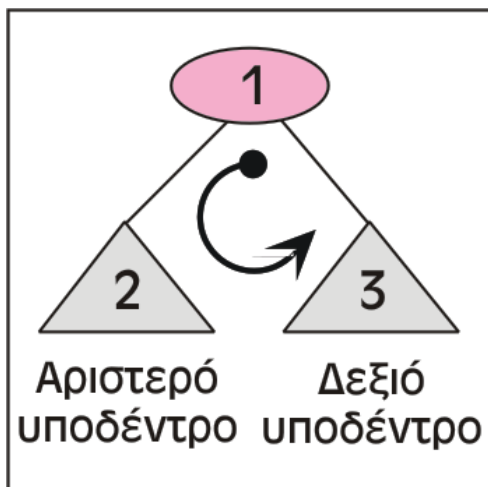
Δυαδικά δένδρα (α) πλήρες, (β) και (γ) μη πλήρη

ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (4)

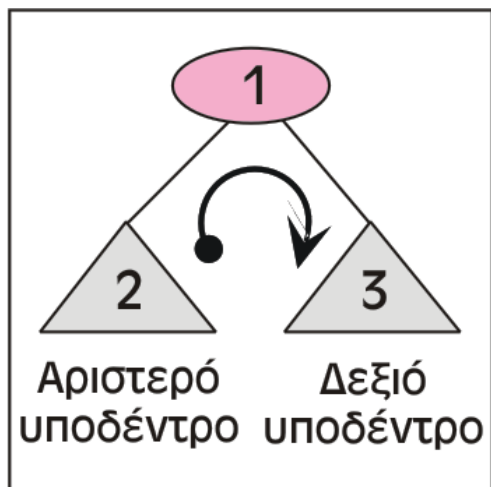
- Κάθε διαδικασία επίσκεψης όλων των κόμβων ενός δένδρου, ακριβώς μια φορά τον καθένα, ονομάζεται **διαπέραση ή διάσχιση (traversal)**
 - **Προδιατεταγμένη** διαπέραση (pre-order): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τον ίδιο τον κόμβο, έπειτα τους κόμβους του αριστερού του υποδένδρου και στη συνέχεια τους κόμβους του δεξιού του υποδένδρου
 - **Μεταδιατεταγμένη** διαπέραση (post-order): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τους κόμβους του αριστερού του υποδένδρου, έπειτα τους κόμβους του δεξιού του υποδένδρου και στη συνέχεια τον ίδιο τον κόμβο
 - **Ενδοδιατεταγμένη** διαπέραση (in-order): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τους κόμβους του αριστερού του υποδένδρου, έπειτα τον ίδιο τον κόμβο και στη συνέχεια τους κόμβους του δεξιού του υποδένδρου
 - Κατά σειρά **επιπέδων** διαπέραση (level-order): Διασχίζουμε τα επίπεδα του δένδρου, στη σειρά από πάνω προς τα κάτω, και σε κάθε επίπεδο επισκεπτόμαστε τους κόμβους από αριστερά προς τα δεξιά

ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (5)

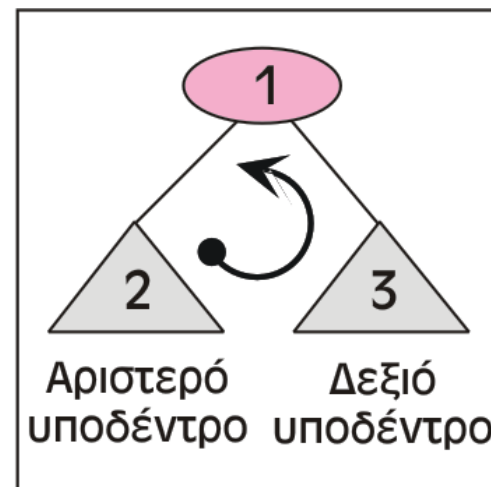
Διάσχιση δυαδικού δέντρου



α. Προδιατεταγμένη
διάσχιση



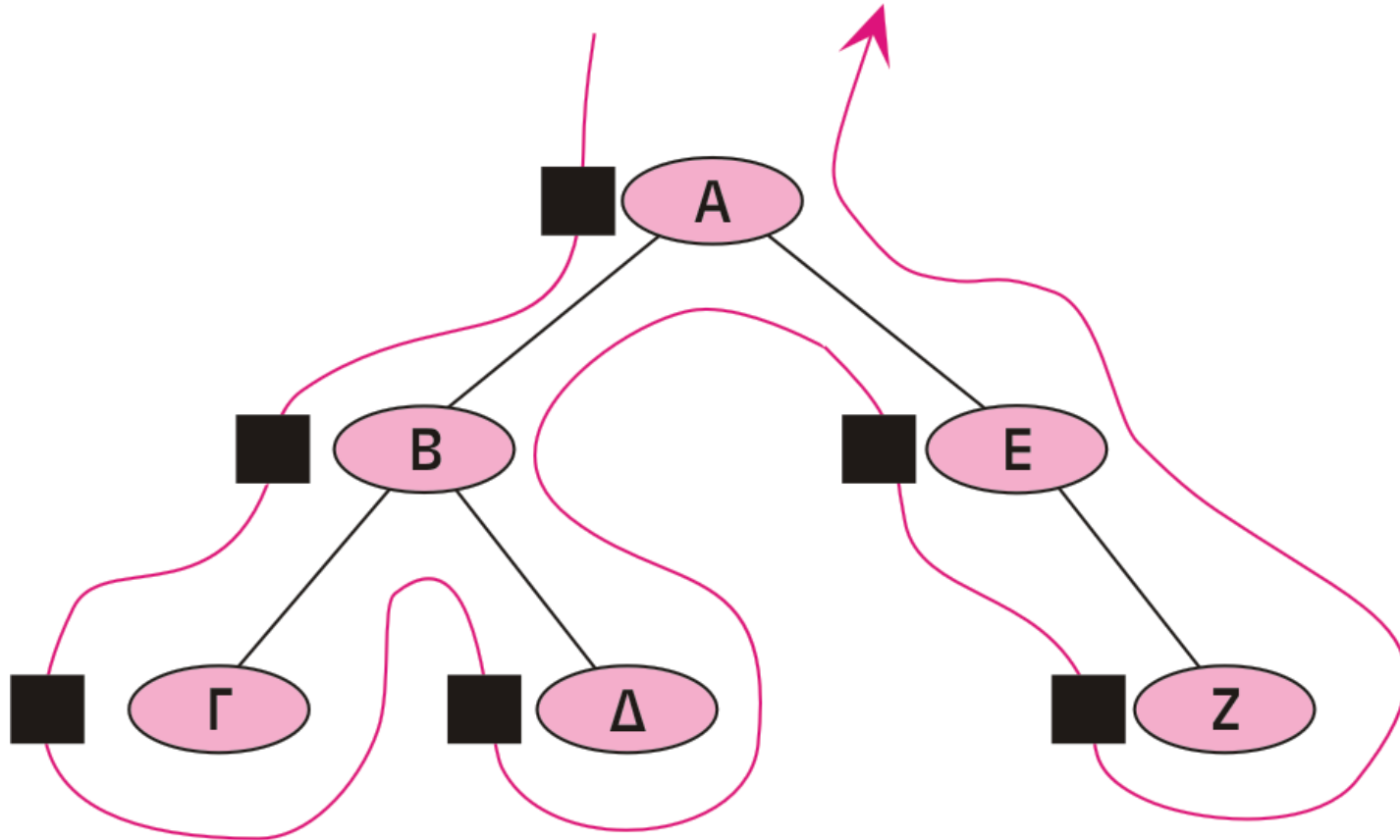
β. Ενδοδιατεταγμένη
διάσχιση



γ. Μεταδιατεταγμένη
διάσχιση

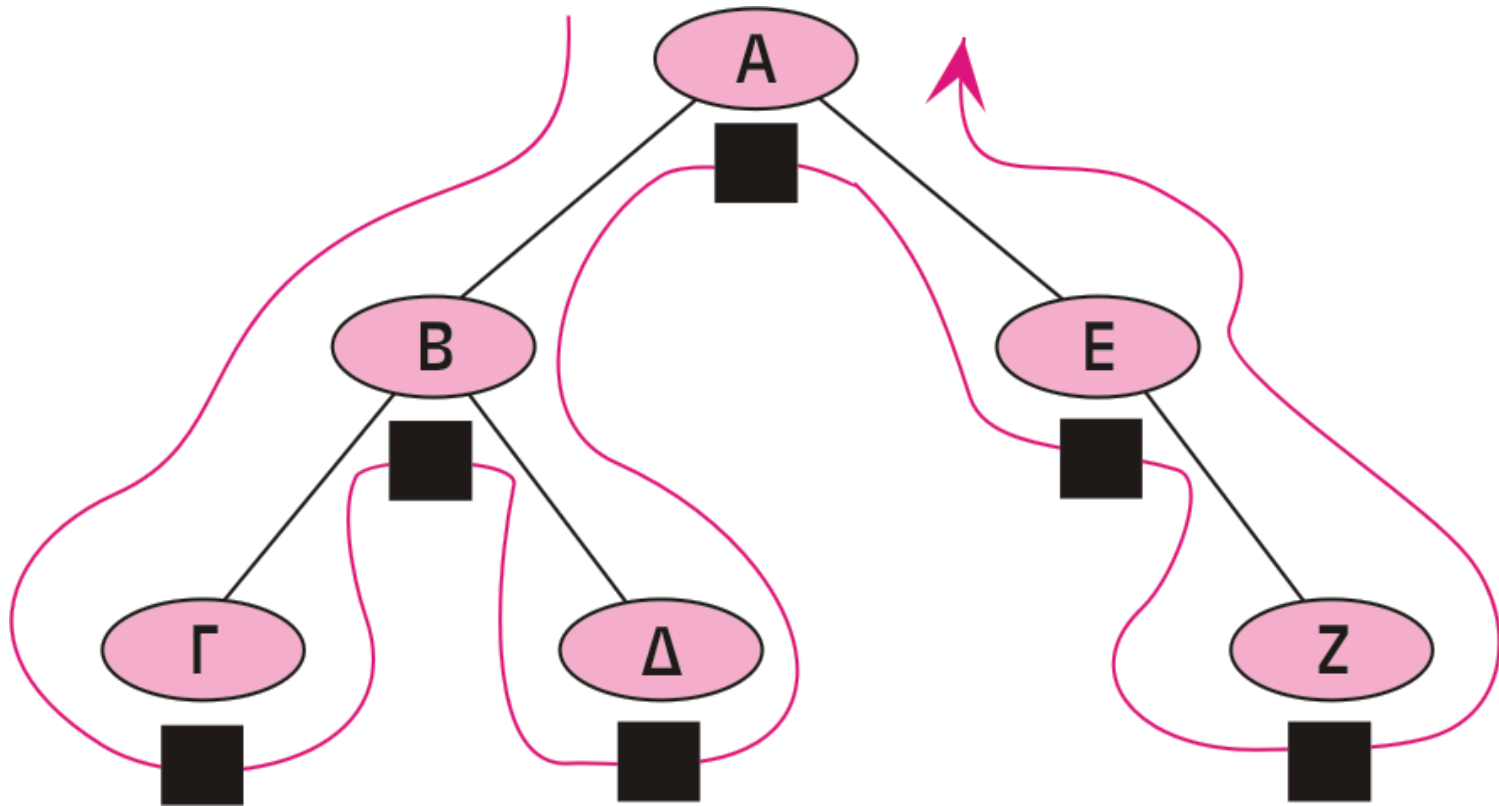
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (6)

Προδιατεταγμένη διάσχιση (preorder traversal)



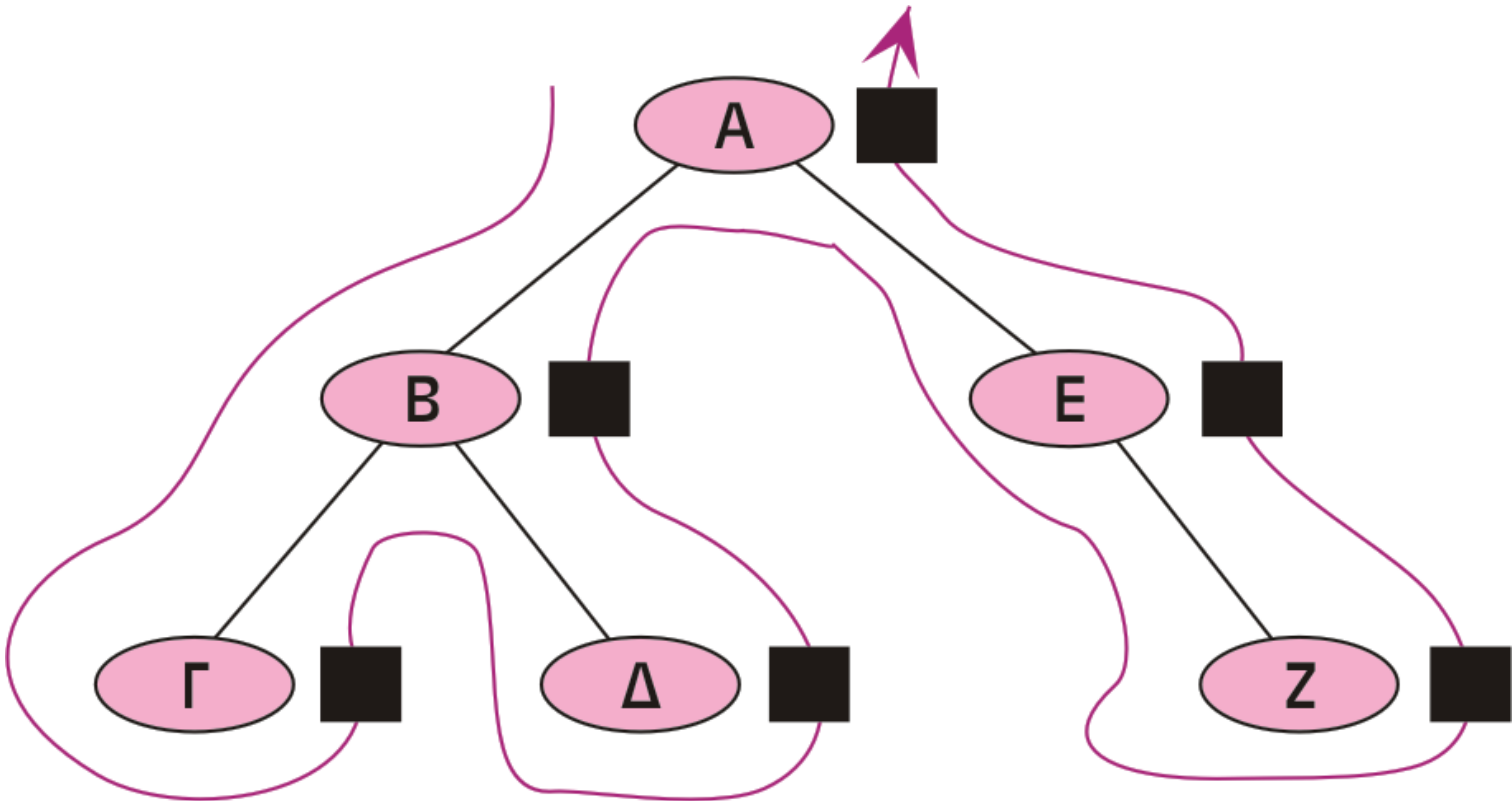
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (7)

Ενδοδιατεταγμένη διάσχιση (inorder traversal)



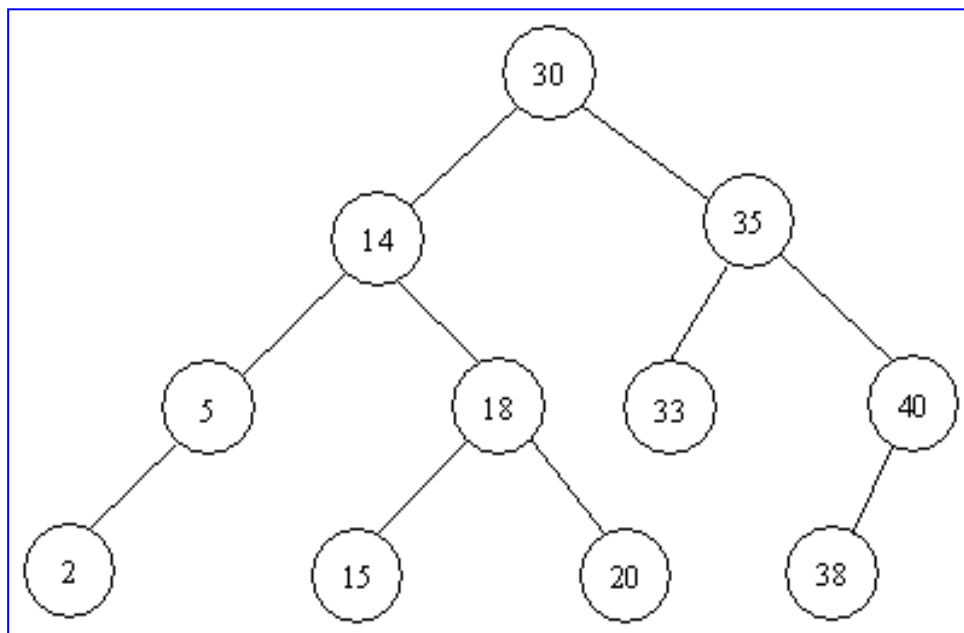
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (8)

Μεταδιατεταγμένη διάσχιση (postorder traversal)



ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ (9)

Θεωρήστε ότι το δέντρο σαρώνεται κατά την προδιατεταγμένη διαπέραση (*preorder traversal*) και τα δεδομένα των κόμβων του εισάγονται σε έναν πίνακα. Γράψτε τον πίνακα μετά τη διαπέραση

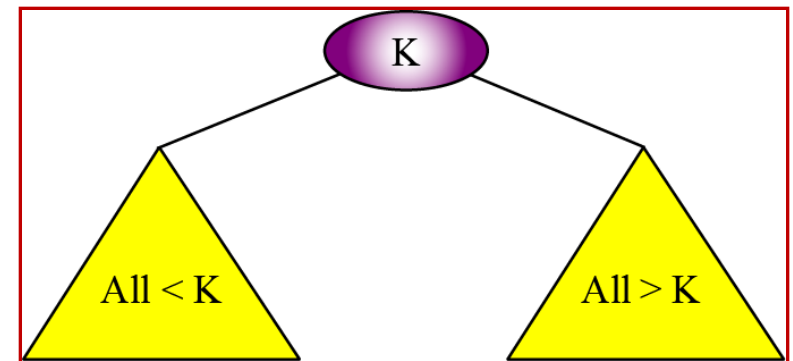
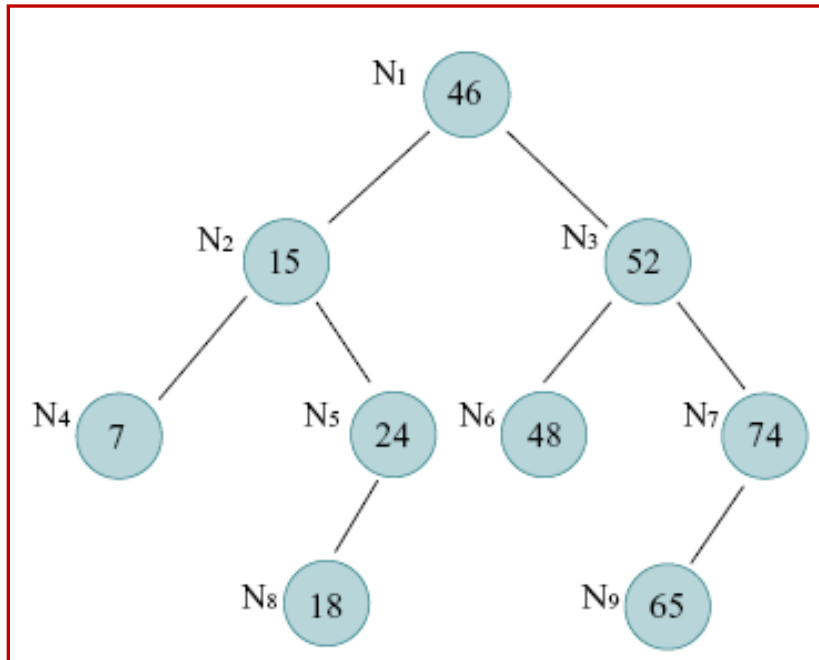


30
14
5
2
18
15
20
35
33
40
38

ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (1)

- Το **αριστερό** υποδένδρο κάθε κόμβου έχει τιμές **μικρότερες** από την τιμή του κόμβου
- Το **δεξί** υποδένδρο κάθε κόμβου έχει τιμές **μεγαλύτερες** από την τιμή του κόμβου
- Χρησιμοποιούνται στην αναζήτηση στοιχείων, όπου ενδέχεται να βελτιώσουν σημαντικά το χρόνο (αντίστοιχα με τη δυαδική αναζήτηση σε ταξινομημένο πίνακα)
- Με ενδοδιατεταγμένη διαπέραση παίρνουμε τα στοιχεία ταξινομημένα κατά αύξουσα σειρά

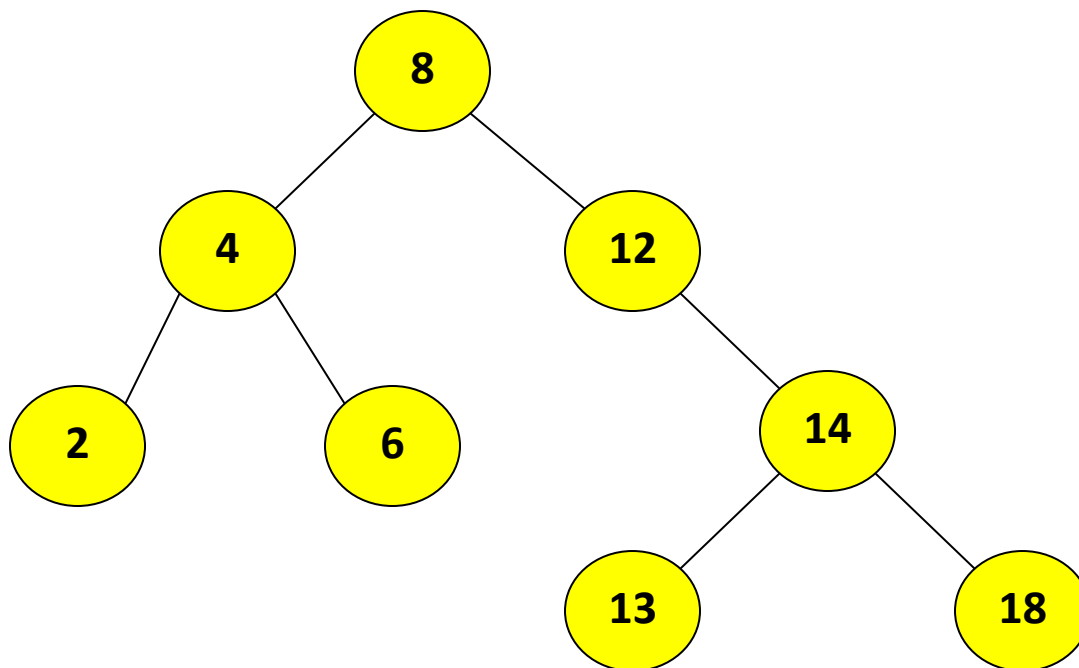
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (2)



Δυαδικό Δένδρο Αναζήτησης – Διαπέραση ΔΔΑ

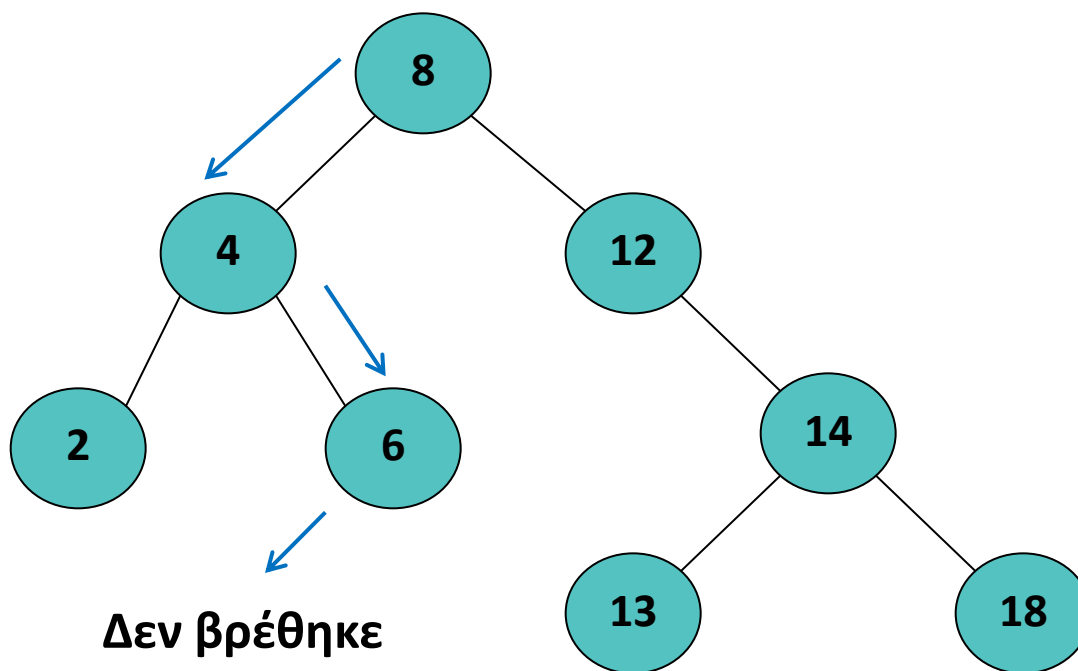
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (3)

Εισαγωγή στοιχείων σε ΔΔΑ: 8, 4, 12, 2, 6, 14, 18, 13



ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (4)

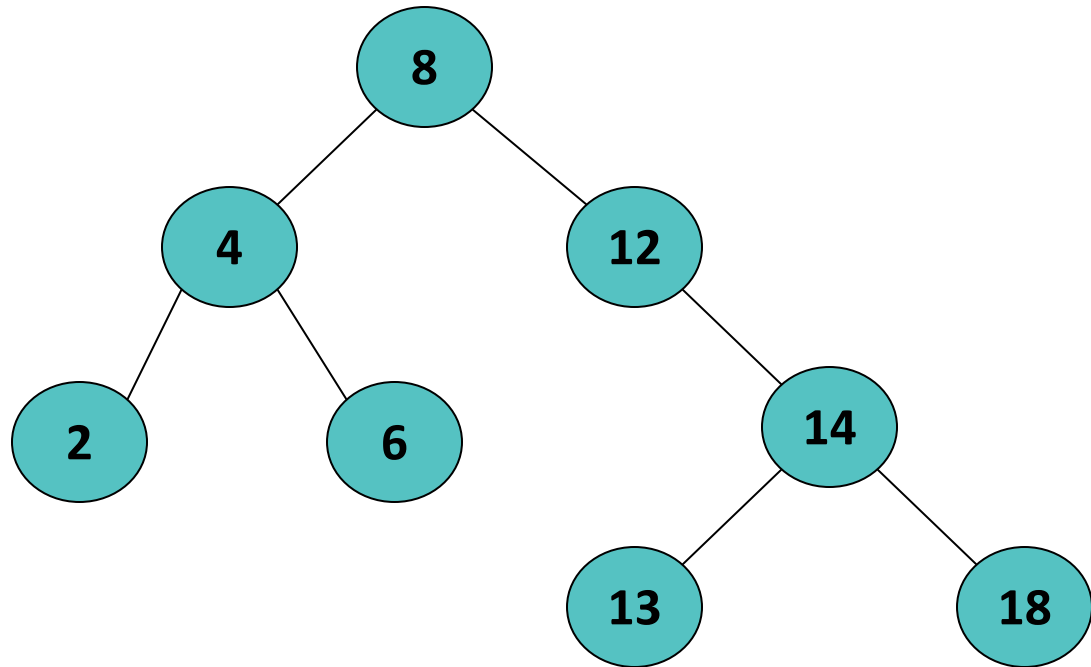
Αναζήτηση του στοιχείου 5



ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (5)

Άσκηση: αναζητήστε ένα προς ένα όλα τα στοιχεία του δένδρου. Καταγράψτε το πλήθος συγκρίσεων μεταξύ των στοιχείων που απαιτούνται.

στοιχείο συγκρίσεις	
8	1
4	2
2	3
6	3
12	2
14	3
13	4
18	4

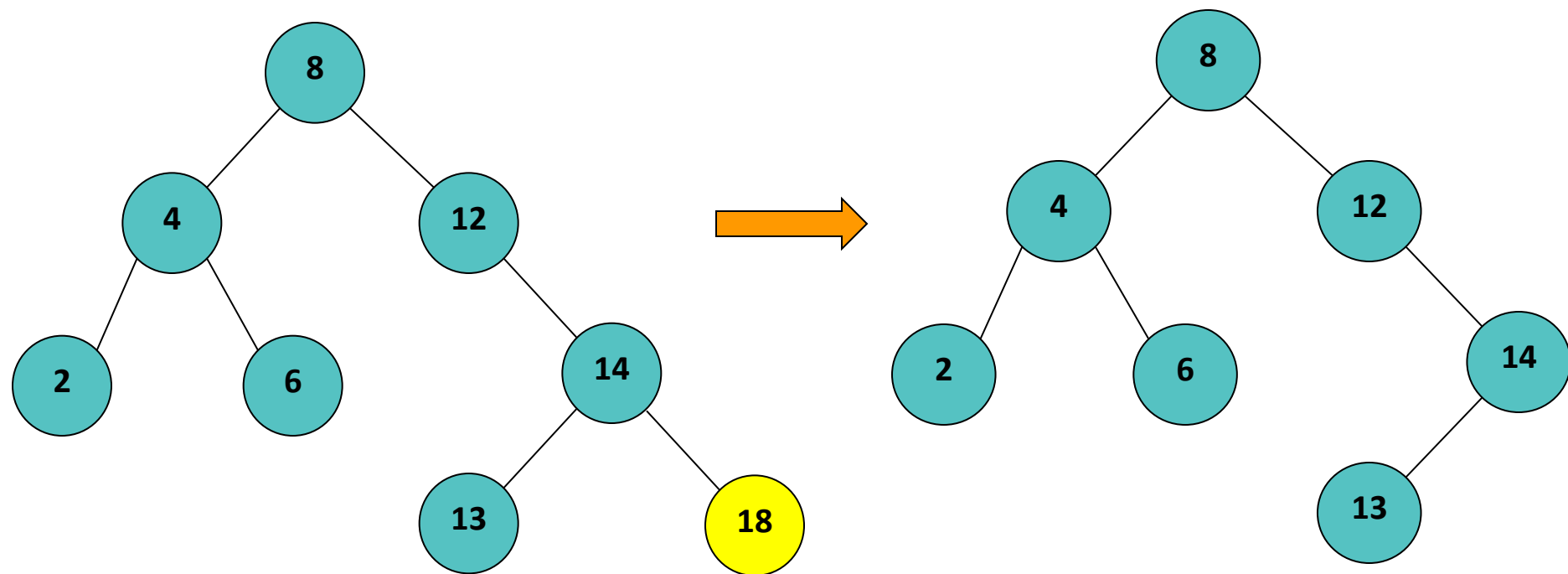


ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (6)

- Διαγραφές
 - Ο κόμβος που διαγράφεται δεν έχει υποδένδρα
 - απλά διαγράφεται από το ΔΔΑ
 - Ο κόμβος που διαγράφεται έχει μόνο ένα υποδένδρο
 - το υποδένδρο αντικαθιστά τον κόμβο που διαγράφεται
 - Ο κόμβος που διαγράφεται έχει δύο υποδένδρα
 - αντικατάσταση με το μεγαλύτερο στοιχείο του αριστερού υποδένδρου ή με το μικρότερο στοιχείο του δεξιού υποδέντρου

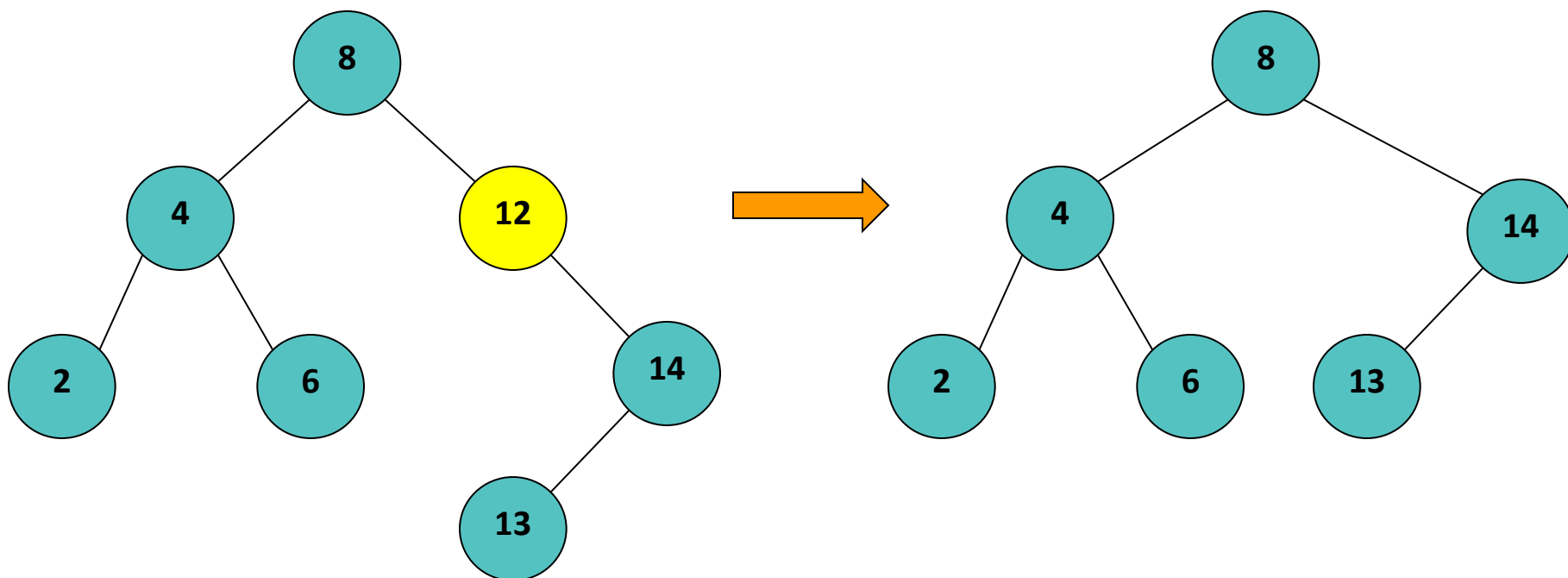
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (7)

Διαγραφή του 18: ο κόμβος που περιέχει το 18 δεν έχει υποδένδρα



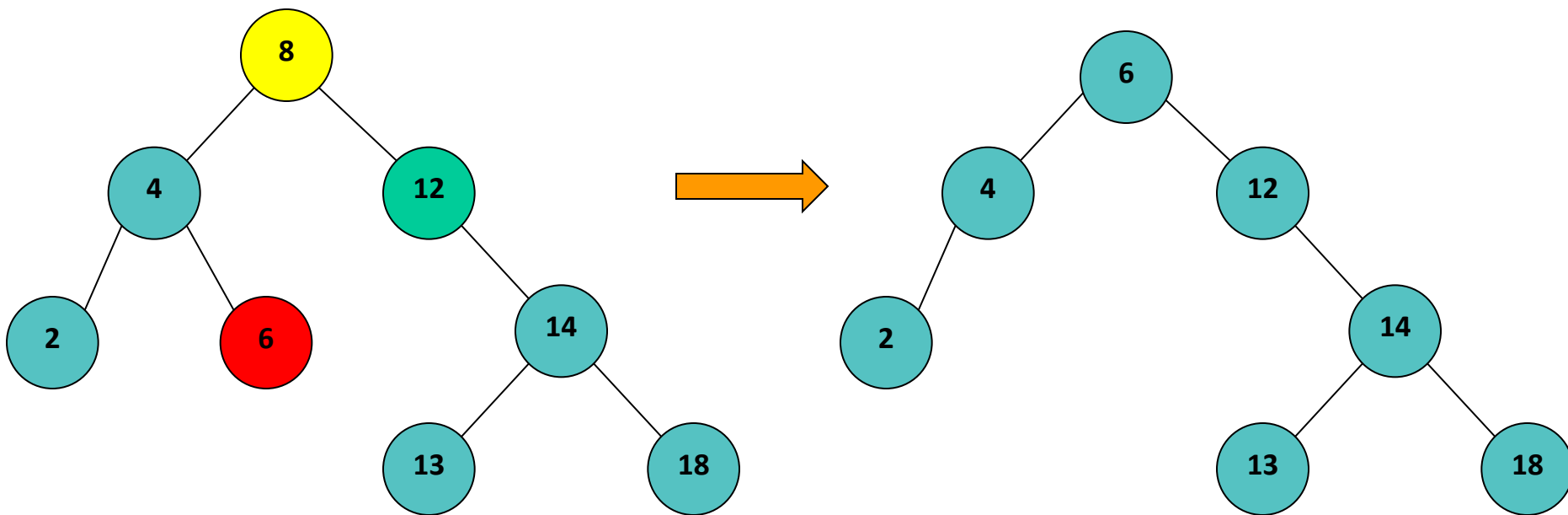
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (8)

Διαγραφή του 12: ο κόμβος που περιέχει το 12 έχει μόνο δεξί υποδένδρο



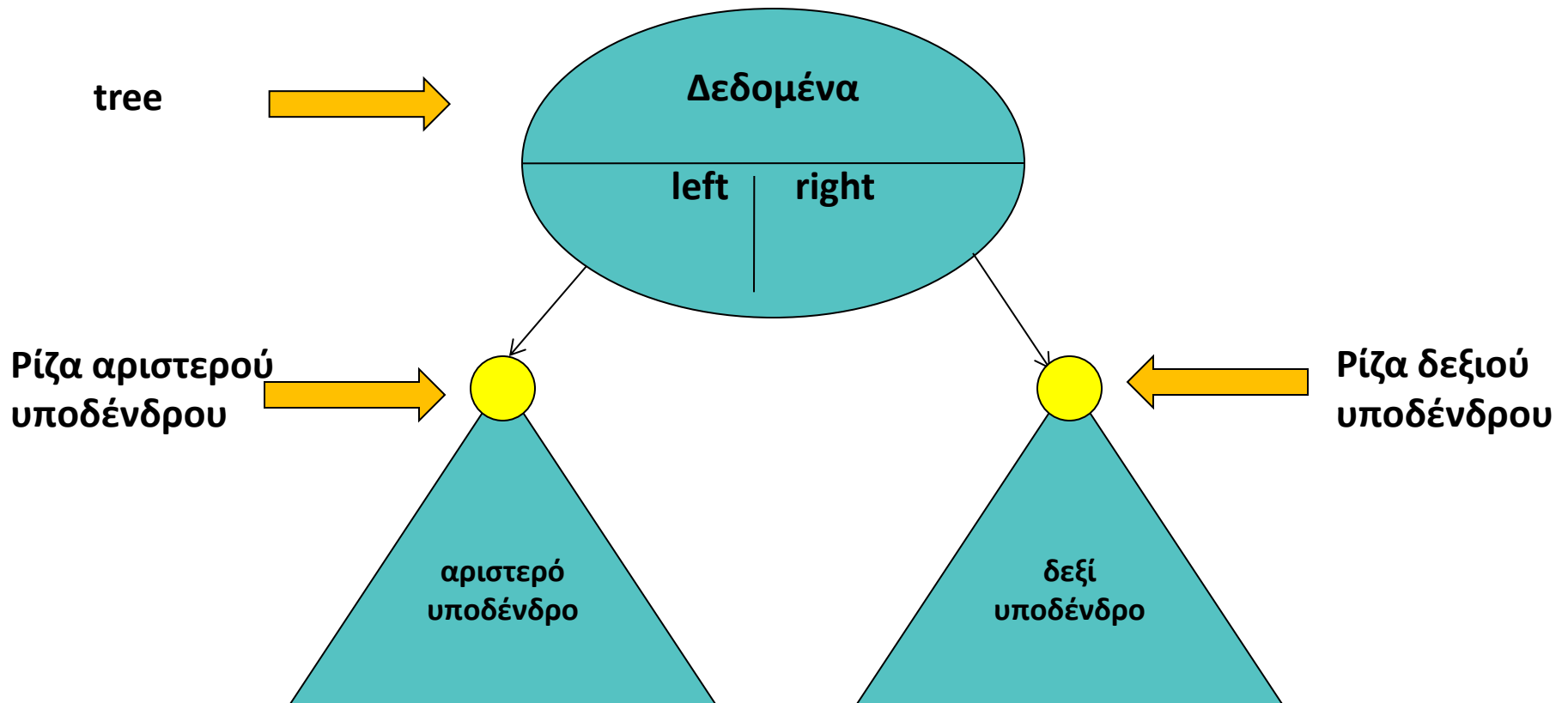
ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ (9)

Διαγραφή του 8. Αντικατάσταση με το μεγαλύτερο του αριστερού υποδένδρου ή με το μικρότερο του δεξιού υποδένδρου



ΥΛΟΠΟΙΗΣΗ

Δομή ενός κόμβου σε συνδεδεμένη αναπαράσταση



ΠΡΑΞΕΙΣ ΓΙΑ ΔΥΑΔΙΚΟ ΔΕΝΔΡΟ ΑΝΑΖΗΤΗΣΗΣ

- Αρχικοποίηση
- Έλεγχος αν είναι κενό
- Εισαγωγή
- Διαγραφή
- Αναζήτηση
- Ενδοδιατεταγμένη διάσχιση
- Μεταδιατεταγμένη διάσχιση
- Προδιατεταγμένη διάσχιση

ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΥΑΔΙΚΟΥ ΔΕΝΤΡΟΥ ΜΕ ΠΙΝΑΚΕΣ

Χρήση πινάκων

- Κάθε κόμβος αποθηκεύεται σε μία θέση πίνακα
- Οι σχέσεις γονέα–παιδιού υλοποιούνται με δείκτες (indices-σε ποια θέση του πίνακα το παιδί)
- Ξεχωριστό πίνακα για *αριστερό* παιδί ξεχωριστό για *δεξί*

key[i] : τιμή κόμβου i
left[i] : δείκτης αριστερού παιδιού
right[i] : δείκτης δεξιού παιδιού
root : δείκτης ρίζας

• -1 $\Rightarrow$ δεν υπάρχει παιδί

ΕΙΣΑΓΩΓΗ (INSERT) ΣΤΟ ΔΕΝΤΡΟ

Insert

1. Είσοδος: x
2. $\text{parent} \leftarrow -1$
3. $\text{current} \leftarrow \text{root}$
4. Όσο $\text{current} \neq -1$ επανάλαβε
 1. $\text{parent} \leftarrow \text{current}$
 2. Αν $x < \text{key}[\text{current}]$ τότε
 1. $\text{current} \leftarrow \text{left}[\text{current}]$
 3. Αλλιώς
 1. $\text{current} \leftarrow \text{right}[\text{current}]$
5. $\text{new} \leftarrow (\text{μέγεθος key}) + 1$
6. $\text{key}[\text{new}] \leftarrow x$
7. $\text{left}[\text{new}] \leftarrow -1$
8. $\text{right}[\text{new}] \leftarrow -1$
9. Αν $\text{parent} = -1$ τότε
 1. $\text{root} \leftarrow \text{new}$
10. Αλλιώς αν $x < \text{key}[\text{parent}]$ τότε
 1. $\text{left}[\text{parent}] \leftarrow \text{new}$
11. Αλλιώς
 1. $\text{right}[\text{parent}] \leftarrow \text{new}$
12. Επιστροφή key , left , right

ΔΙΑΓΡΑΦΗ (DELETE) ΣΤΟ ΔΕΝΤΡΟ

- Η διαγραφή ξεκινά όπως η αναζήτηση, αλλά κρατάμε και τον κόμβο γονέα (parent).

Συνάρτηση Delete

1. Είσοδος: x
2. $\text{parent} \leftarrow -1$
3. $\text{current} \leftarrow \text{root}$
4. Όσο $\text{current} \neq -1$ και $\text{key}[\text{current}] \neq x$ επανάλαβε
 1. $\text{parent} \leftarrow \text{current}$
 2. Αν $x < \text{key}[\text{current}]$ τότε
 1. $\text{current} \leftarrow \text{left}[\text{current}]$
 3. Αλλιώς
 1. $\text{current} \leftarrow \text{right}[\text{current}]$
5. Αν $\text{current} = -1$ τότε
 1. Επιστροφή -1

Αν δεν βρεθεί κόμβος, επιστρέφουμε. Αν βρεθεί όμως, τι κάνουμε;

ΔΙΑΓΡΑΦΗ (DELETE) ΣΤΟ ΔΕΝΤΡΟ

Περίπτωση 1: κανένα παιδί (ο κόμβος είναι φύλλο)

6. Αν $\text{left}[\text{current}] = -1$ και $\text{right}[\text{current}] = -1$ τότε
 1. Αν $\text{parent} = -1$ τότε
 1. $\text{root} \leftarrow -1$
 2. Αλλιώς αν $\text{left}[\text{parent}] = \text{current}$ τότε
 1. $\text{left}[\text{parent}] \leftarrow -1$
 3. Αλλιώς
 1. $\text{right}[\text{parent}] \leftarrow -1$

Περίπτωση 2: ο κόμβος έχει 1 παιδί

7. Αλλιώς Αν $\text{left}[\text{current}] = -1$ ή $\text{right}[\text{current}] = -1$ τότε
 7. $\text{child} \leftarrow$ το μοναδικό παιδί του current
 8. Αν $\text{parent} = -1$ τότε
 7. $\text{root} \leftarrow \text{child}$
 9. Αλλιώς αν $\text{left}[\text{parent}] = \text{current}$ τότε
 7. $\text{left}[\text{parent}] \leftarrow \text{child}$
 10. Αλλιώς
 7. $\text{right}[\text{parent}] \leftarrow \text{child}$

ΔΙΑΓΡΑΦΗ (DELETE) ΣΤΟ ΔΕΝΤΡΟ

Υλοποίηση βάζοντας στην θέση του κόμβου το **μικρότερο στοιχείο στο δεξί υποδέντρο**

Περίπτωση 3: Ο κόμβος έχει 2 παιδιά

8. Αλλιώς

1. $\text{successor_parent} \leftarrow \text{current}$
2. $\text{successor} \leftarrow \text{right}[\text{current}]$
3. Όσο $\text{left}[\text{successor}] \neq -1$ επανάλαβε
 8. $\text{successor_parent} \leftarrow \text{successor}$
 9. $\text{successor} \leftarrow \text{left}[\text{successor}]$
4. $\text{key}[\text{current}] \leftarrow \text{key}[\text{successor}]$
5. Αν $\text{left}[\text{successor_parent}] = \text{successor}$ τότε
 8. $\text{left}[\text{successor_parent}] \leftarrow \text{right}[\text{successor}]$
6. Αλλιώς
 8. $\text{right}[\text{successor_parent}] \leftarrow \text{right}[\text{successor}]$

9. Επιστροφή 1

BST ANIMATION

- <https://yongdanielliang.github.io/animation/web/BST.html>