Κεφάλαιο 10

Δένδρα

Όλες οι προηγούμενες δομές που εξετάστηκαν, λέγονται ***γραμμικές*** (*linear*), διότι διαθέτουν τη σχέση της συνεχόμενης λογικής γειτονικότητας των στοιχείων τους. Στο κεφάλαιο αυτό θα εξετάσουμε τα δένδρα, που είναι μια πολύ σπουδαία μη γραμμική δομή. Η δομή του δένδρου σε γενικές γραμμές δεν είναι τίποτε άλλο παρά η σύνδεση κόμβων με τρόπο ανάλογο ενός πραγματικού δένδρου.

Ένας ορισμός των δένδρων είναι ο ακόλουθος:

Δένδρο **(*tree*)** είναι ένα πεπερασμένο σύνολο κόμβων που συνδέονται με κατευθυνόμενες ακμές. Υπάρχει ένας μοναδικός κόμβος που ονομάζεται ***ρίζα*** **(*root*)** και στον οποίο δεν καταλήγουν αλλά μόνο ξεκινούν ακμές. Από κάθε κόμβο μπορούν να ξεκινούν καμία, μία ή περισσότερες ακμές. Σε κάθε κόμβο, εκτός της ρίζας, καταλήγει ακριβώς μία ακμή. Για οποιοδήποτε κόμβο του δένδρου, εκτός της ρίζας, υπάρχει μία μοναδική ακολουθία ακμών που οδηγεί από τη ρίζα στον κόμβο.

Συχνά δίνεται και ένας, λιγότερο διαισθητικός, αναδρομικός ορισμός. Σύμφωνα με αυτόν, ένα δένδρο είναι είτε κενό είτε αποτελείται από ένα πεπερασμένο σύνολο κόμβων, από τους οποίους ένας θεωρείται ***ρίζα*** και οι υπόλοιποι είναι μοιρασμένοι σε καμία, μία ή περισσότερες ξένες ομάδες κόμβων. Κάθε ομάδα αποτελεί ένα δένδρο ενώ υπάρχει ακριβώς μία ακμή από τη ρίζα του δένδρου προς ένα μοναδικό κόμβο κάθε ομάδας κόμβων που θεωρείται ρίζα του εκάστοτε υποδένδρου. Αξίζει

να παρατηρήσουμε ότι για κάθε υποδένδρο, στο επόμενο επίπεδο της αναδρομής, ορίζονται πιθανώς νέα εσωτερικά υποδένδρα και ακμές που συνδέουν τη ρίζα του με τις ρίζες αυτών των εσωτερικών υποδένδρων, κ.ο.κ. Θα επανέλθουμε σε αυτόν τον ορισμό πιο τυπικά όταν θα ασχοληθούμε με τη διάσχιση δένδρων.

Στο σχήμα 10.1 φαίνονται 4 διαφορετικές αναπαραστάσεις δένδρων. Αυτή που χρησιμοποιείται συνήθως γιατί δίνει καλύτερη εποπτεία των συσχετίσεων των δεδομένων (κόμβων) είναι η μορφή α, του ανεστραμμένου δένδρου με τη ρίζα στην κορυφή. Η μορφή αυτή θα χρησιμοποιείται από δω και στο εξής. Η *φορά* των ακμών είναι από πάνω προς τα κάτω και υπονοείται, για αυτό και συχνά δεν απεικονίζεται.

Το δένδρο αποτελεί μία κατ' εξοχήν ***ιεραρχική*** δομή. Η ορολογία του προκύπτει τόσο από τα φυσικά δένδρα (ρίζα, φύλλο, κλπ.) όσο και από τα γενεαλογικά δένδρα (γονέας, παιδί, κ.λπ.).

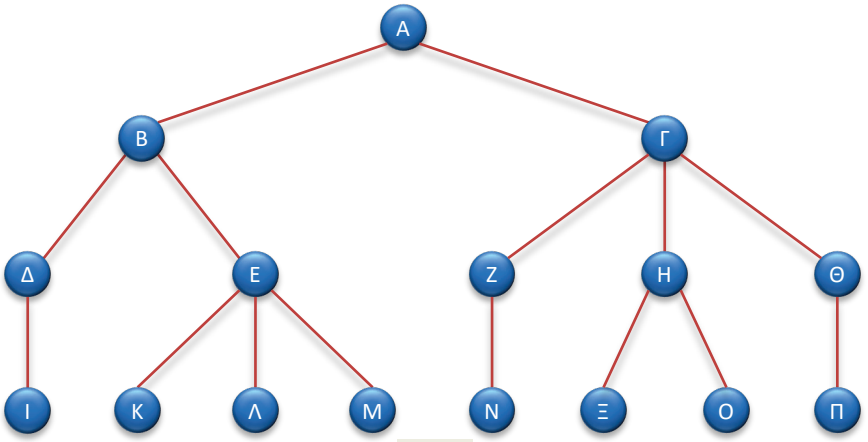
10.1 Ορισμοί

Ένας κόμβος α είναι ο (μοναδικός) ***γονέας*** (*parent*) του κόμβου β, όταν από τον α ξεκινά η ακμή που καταλήγει στον β, ενώ ο β θεωρείται ***παιδί*** (*child*) του α. Όπως προκύπτει από τον ορισμό του δένδρου, ακολουθώντας αυτήν την ορολογία, οποιοσδήποτε κόμβος (εκτός από τη ρίζα) έχει ***ακριβώς έναν γονέα*** αλλά μπορεί να διαθέτει ***κανένα, ένα*** *ή* ***περισσότερα παιδιά***. Η ορολογία αυτή επεκτείνεται προς τα πάνω ή προς τα κάτω γενικότερα, με όρους που σχετίζονται με ***προγόνους*** (*ascendants*) και ***απογόνους*** (*descendants*).

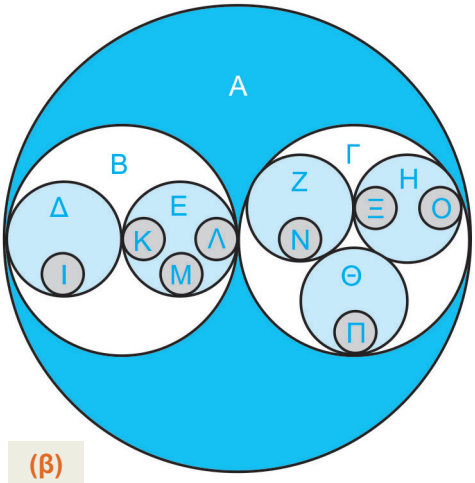
Για παράδειγμα στο σχήμα 10.1α ο κόμβος Γ είναι ο γονέας των Ζ, Η, Θ και αυτοί είναι παιδιά του Γ. Ο Γ είναι πρόγονος των Ν, Ξ, Ο, Π και οι τελευταίοι είναι απόγονοι του Γ. Συχνά, όταν δύο κόμβοι έχουν κοινό γονέα, αναφέρονται και ως ***αδέλφια*** ή ***αμφιθαλείς*** (*siblings*). Στο ίδιο σχήμα, οι κόμβοι Κ, Λ και Μ είναι αδέλφια, αφού έχουν κοινό γονέα τον Ε.

Βαθμός (*degree*) ενός *κόμβου* είναι ο αριθμός των παιδιών του. ***Βαθμός*** ενός ***δένδρου*** είναι ο μέγιστος από τους βαθμούς των κόμβων του. Δένδρα ***βαθμού d*** ονομάζονται ***d-αδικά*** (*d-ary*) δένδρα.

Συχνά, ιδιαίτερα σε επίπεδο υλοποίησης, μπορεί να εννοείται ως "βαθμός" ενός δένδρου ο *μέγιστος επιτρεπόμενος βαθμός* για οποιοδήποτε κόμβο του, ακόμα και αν, σε ένα συγκεκριμένο στιγμιότυπο του δένδρου, δεν υπάρχει κόμβος αυτού του βαθμού. Αντίστοιχα, με το ίδιο σκεπτικό, "βαθμός" ενός κόμβου μπορεί να θεωρείται το μέγιστο επιτρεπόμενο πλήθος παιδιών ενός κόμβου.



(α)



(β)

(γ)

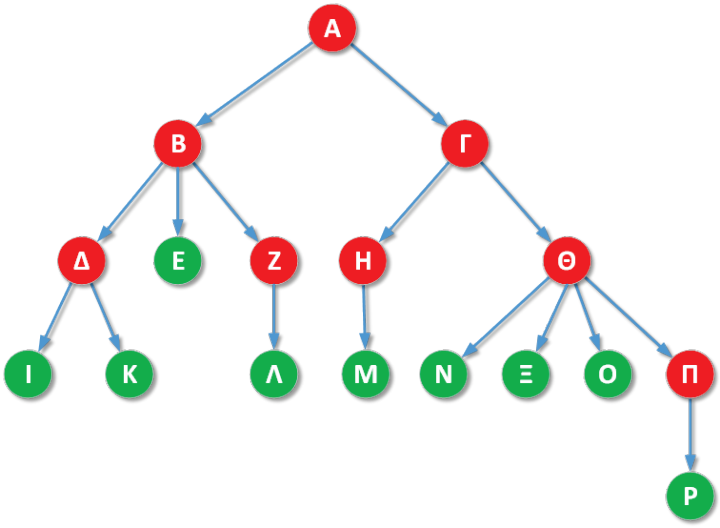
(δ) (**Α(** Β(Δ(Ι), Ε(Κ, Λ, Μ) **)**, **Γ(** Ζ(Ν), Η(Ξ, Ο), Θ(Π) **)** **)**

Σχήμα 10.1. *Τέσσερις διαφορετικές αναπαραστάσεις του ίδιου δένδρου*

Στο σχήμα 10.1α ο ***βαθμός*** του ***κόμβου Β*** είναι ***2***, γιατί έχει δύο παιδιά. Το ***δένδρο*** όμως είναι ***βαθμού 3***, γιατί ο μεγαλύτερος βαθμός κόμβου είναι 3 (των Γ και Ε). Αντίστοιχα, σε επίπεδο υλοποίησης, το συγκεκριμένο δένδρο μπορεί να είναι στιγμιότυπο οποιουδήποτε d-αδικού δένδρου με $d \geq 3$.

Τα δένδρα που χρησιμοποιούνται συνήθως στην πράξη είναι ***δυαδικά*** *(binary)*, ***τριαδικά*** *(ternary)* ή ***τετραδικά*** *(quadtree)*. Ωστόσο, για τις ανάγκες της μόνιμης αποθήκευσης, αξιοποιούνται και δένδρα μεγαλύτερου βαθμού.

Οι κόμβοι που δεν έχουν παιδιά λέγονται ***εξωτερικοί κόμβοι*** (*external nodes*) ή ***τερματικοί κόμβοι*** (*terminal nodes*) ή ***φύλλα*** (*leaves*). Οι υπόλοιποι κόμβοι λέγονται ***εσωτερικοί κόμβοι*** (*internal nodes*) ή ***μη τερματικοί κόμβοι*** (*non-terminal nodes*).



Σχήμα 10.2. *Τερματικοί/εξωτερικοί κόμβοι ή φύλλα (πράσινο χρώμα) και μη-τερματικοί/εξωτερικοί κόμβοι (κόκκινο χρώμα)*

Επίπεδο (*level*) ή ***βάθος*** (*depth*) ενός κόμβου είναι το πλήθος των προγόνων του (μέχρι και τη ρίζα). Για παράδειγμα, στο σχήμα 10.1α ο κόμβος Δ βρίσκεται στο επίπεδο 2 ή έχει βάθος 2. Η ρίζα θεωρείται ότι βρίσκεται στο επίπεδο 0 και έχει βάθος 0.[1] ***Ύψος*** (*height*) ενός κόμβου θεωρείται το μέγιστο πλήθος ακμών μέχρι κάποιο φύλλο. Για παράδειγμα, στο δένδρο του Σχήματος 10.2, το ύψος του κόμβου Β είναι 2 ενώ το ύψος του κόμβου Γ είναι 3, παρότι είναι αδέλφια.

Το μέγιστο επίπεδο των κόμβων ενός δένδρου ή, εναλλακτικά, το ύψος της ρίζας, λέγεται ***βάθος*** (depth) ή ***ύψος*** *(height)* του δένδρου. Το δένδρο του σχήματος 10.1α

[1,2] Στη βιβλιογραφία συναντάμε και μία εναλλακτική προσέγγιση αρίθμησης των επιπέδων και συνακόλουθης καταμέτρησης του ύψους, που θεωρεί το επίπεδο οποιουδήποτε κόμβου ως το πλήθος των προγόνων του αυξημένο κατά 1. Σύμφωνα με αυτήν την προσέγγιση, το επίπεδο της ρίζας είναι 1 και το δένδρο του Σχήματος 10.1α είναι ύψους 4. Ωστόσο, η εκδοχή κατά την οποία το επίπεδο της ρίζας θεωρείται 0 αποτελεί την πιο διαδεδομένη και αυτή ακολουθείται στο παρόν.

έχει ύψος 3 ενώ αυτό του σχήματος 10.2 έχει ύψος 4.[2] Σύμφωνα με αυτήν τη θεώρηση, το ύψος ενός δένδρου που περιέχει μόνο τη ρίζα θεωρείται 0 ενώ το ύψος ενός κενού δένδρου θεωρείται –1.

Υποδένδρο ονομάζεται το δένδρο που σχηματίζεται, αν ως ρίζα ληφθεί ένας οποιοσδήποτε κόμβος.

Ως ***μονοπάτι*** (path) μεταξύ των κόμβων α και β σε ένα δένδρο (αν υπάρχει) ορίζεται η ακολουθία κόμβων και ακμών του δένδρου που οδηγεί από τον α στον β. Ως ***μήκος μονοπατιού*** (path length) ορίζεται το πλήθος των ακμών στο μονοπάτι.

Σύμφωνα με τον ορισμό του δένδρου, για οποιοδήποτε κόμβο του εκτός της ρίζας υπάρχει ακριβώς ένα μονοπάτι που οδηγεί από τη ρίζα στον κόμβο. Από τα παραπάνω προκύπτει ότι ***ένα δένδρο δεν περιέχει κύκλους*****,** δηλαδή δεν μπορούμε να ξεκινήσουμε και να καταλήξουμε στον ίδιο κόμβο ακολουθώντας ακμές του δένδρου.

Υπάρχουν επίσης κάποιες ***συμβάσεις απεικόνισης*** που ακολουθούνται και ήδη χρησιμοποιήσαμε στην αναπαράσταση των δένδρων στα προηγούμενα σχήματα. Η πρώτη αναφέρθηκε ήδη και αφορά την τοποθέτηση της ρίζας στην κορυφή της γραφικής αναπαράστασής του, ενώ οι κόμβοι ίδιου επιπέδου/βάθους απεικονίζονται όντως στο ίδιο οριζόντιο επίπεδο. Ως αποτέλεσμα αυτών η κατεύθυνση των ακμών είναι ευνόητη και συνήθως παραλείπεται. Επιπλέον, αποφεύγουμε τη διασταύρωση ακμών στην απεικόνιση ενός δένδρου. Τέλος, συνηθίζεται το περιεχόμενο που απεικονίζεται σε κάθε κόμβο να είναι μόνο η πληροφορία που αφορά το εκάστοτε πρόβλημα, ακόμα και αν ο κόμβος περιέχει στην πράξη και επιπλέον πληροφορίες (είτε δεδομένων, είτε διαχειριστικές, διευθύνσεων μνήμης κλπ.).

10.2 Ποσοτικά στοιχεία

Ένα δένδρο που κάθε εσωτερικός κόμβος του έχει το μέγιστο πλήθος παιδιών (σύμφωνα με το βαθμό του δένδρου) λέγεται ***πλήρες*** (full). Ένα πλήρες δένδρο που όλα του τα φύλλα βρίσκονται στο ίδιο επίπεδο λέγεται ***τέλειο*** (perfect).[3] Παρατηρήστε τη διαφοροποίηση στο παρακάτω σχήμα όπου, παρότι κάθε εσωτερικός κόμβος έχει το μέγιστο πλήθος παιδιών στο πλήρες (full) δένδρο (Σχ. 10.3α), δεν βρίσκονται όλα τα φύλλα του στο ίδιο επίπεδο όπως συμβαίνει στο τέλειο (perfect) δένδρο (Σχ. 10.3β). Υπάρχουν περιπτώσεις στη βιβλιογραφία που με τον όρο πλήρες

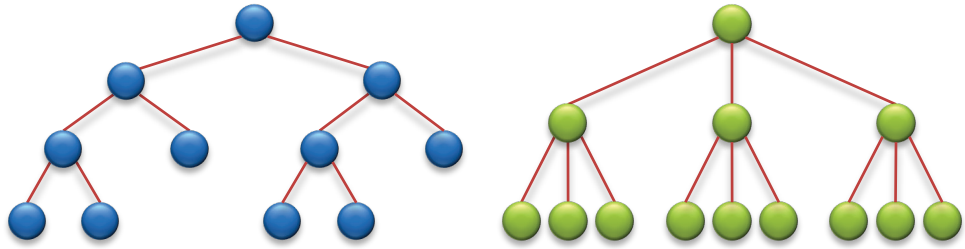
[3] Θα δούμε αργότερα και την έννοια του *συμπληρωμένου* (complete) δένδρου που όμως σχετίζεται με τη διάταξη των παιδιών ενός κόμβου – προς το παρόν τα παιδιά ενός κόμβου θεωρούνται μη διατεταγμένα.

n-αδικό δένδρο ύψους h εννοείται το τέλειο n-αδικό δένδρο ύψους h, ωστόσο εδώ θα τηρήσουμε τη διαφοροποίηση.

Ένα δένδρο βαθμού d και ύψους h μπορεί να έχει το πολύ

$$n = \sum_{i=0}^{h} d^i = 1 + d + d^2 + \cdots + d^h = \frac{d^{h+1} - 1}{d - 1}, \qquad d \neq 1$$

κόμβους, αντιστοιχώντας, πρακτικά, στο τέλειο δένδρο βαθμού d και ύψους h.



(α) **(β)**

Σχήμα 10.3. (α) *Ένα πλήρες (full) δυαδικό δένδρο ύψους 3 και* ***(β)*** *ένα τέλειο (perfect) τριαδικό δένδρο ύψους 2*

Με εφαρμογή του προηγούμενου τύπου προκύπτει ότι ο αριθμός των κόμβων ενός τέλειου δυαδικού (d = 2) δένδρου, ύψους h είναι $2^{h+1} - 1$. Άμεση συνέπεια του προηγουμένου είναι, ότι ένα τέλειο δυαδικό δένδρο με n κόμβους έχει ύψος log(n+1)–1.

Μήκος μονοπατιού ενός κόμβου λέγεται ο αριθμός των ακμών που πρέπει να διασχίσουμε για να φτάσουμε από τη ρίζα σε αυτόν τον κόμβο, πρακτικά αντιστοιχώντας στο βάθος του κόμβου. Με βάση αυτόν τον ορισμό, το μήκος μονοπατιού μέχρι τη ρίζα είναι 0.

Εσωτερικό μήκος μονοπατιού ενός δένδρου είναι το άθροισμα του μήκους των μονοπατιών όλων των κόμβων του δένδρου (για πληρότητα έχουμε συμπεριλάβει το μηδενικό μονοπάτι μέχρι τη ρίζα στο άθροισμα):

$$I = \sum_{i=0}^{h} i \cdot n_i$$

όπου ***n*** το πλήθος των κόμβων στο (μη κενό) δένδρο, ***h*** το ύψος του και $\boldsymbol{n_i}$ το πλήθος των κόμβων στο επίπεδο ***i***.

Μέσο εσωτερικό μήκος μονοπατιού είναι το πηλίκο του εσωτερικού μήκους μονοπατιού δια το πλήθος των κόμβων και δίνεται από τον αντίστοιχο τύπο:

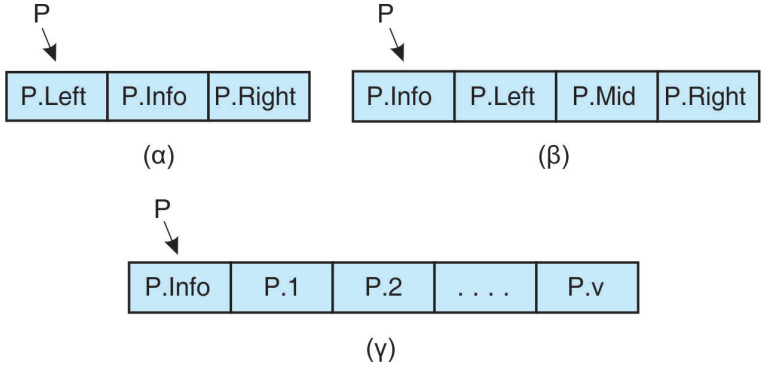
$$I_M = \frac{I}{n} = \frac{1}{n}\sum_{i=0}^{h} i \cdot n_i\,, \qquad n \neq 0$$

Εξωτερικό μήκος μονοπατιού ενός δένδρου είναι το άθροισμα του μήκους των μονοπατιών προς όλους τους κενούς κόμβους του δένδρου, δηλαδή προς τους κενούς δείκτες των τερματικών του κόμβων. Μπορούμε εύκολα να δούμε ότι το εσωτερικό μήκος μονοπατιού σε ένα δένδρο με μοναδικό κόμβο τη ρίζα είναι 0 και το εξωτερικό μήκος μονοπατιού είναι 2. Στη συνέχεια, μπορεί σχετικά εύκολα να δειχθεί ότι κάθε νέος κόμβος του δένδρου προσαυξάνει κατά 2 στο εξωτερικό μήκος μονοπατιού σε σχέση με την εκάστοτε νέα τιμή του εσωτερικού μήκους μονοπατιού οπότε έχουμε:

$$E = 2 \cdot n + I = 2 \cdot n + \sum_{i=0}^{h} i \cdot n_i$$

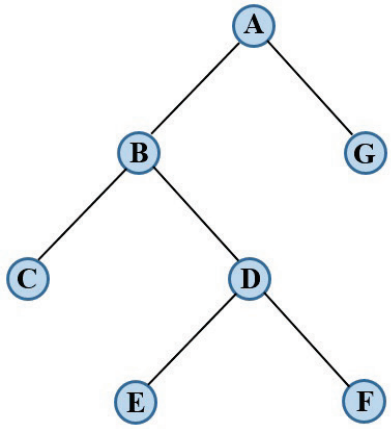
10.3 Αποθήκευση δένδρων

Τα δένδρα χρησιμοποιούνται για την ***ιεραρχική αναπαράσταση δεδομένων*** (*κόμβοι*) και των ***συσχετίσεών*** τους (*ακμές*). Στη μνήμη ενός υπολογιστή μπορεί να χρησιμοποιηθεί ανάλογη τεχνική για την αναπαράστασή τους με αυτήν που χρησιμοποιείται και στις λίστες. Κάθε κόμβος αποτελείται από τα δεδομένα που περιέχει και από ένα σύνολο δεικτών ίσων σε πλήθος με το βαθμό του δένδρου. Οι δείκτες δείχνουν στα παιδιά του κόμβου. Αν δεν υπάρχει κάποιο παιδί, τότε ο αντίστοιχος δείκτης είναι ***κενός*** (NULL).



Σχήμα 10.4. *Τυπικοί κόμβοι* ***(α)*** *δυαδικού δένδρου,* ***(β)*** *τριαδικού δένδρου,* ***(γ)*** *ν-δικού δένδρου.*

Για την αποθήκευση δένδρων μπορούν να χρησιμοποιηθούν και ***πίνακες***. Μία εκδοχή είναι ένας αλφαριθμητικός πίνακας Info που τηρεί το περιεχόμενο των κόμβων, ενώ αριθμητικοί πίνακες ίσοι σε πλήθος με το βαθμό του δένδρου περιέχουν τις θέσεις των παιδιών στον πίνακα Info. Στο σχήμα 10.5β φαίνεται η αποθήκευση σε πίνακες του δυαδικού δένδρου του σχήματος 10.5α. Εναλλακτικά, μπορεί να θεωρηθεί ότι κάθε θέση του πίνακα είναι του κατάλληλου σύνθετου τύπου δεδομένων ώστε να περιέχει τα δεδομένα (info) και τους ακεραίους που υποδεικνύουν τους δείκτες των παιδιών του κόμβου.



	left	info	right
1	2	A	7
2	3	B	4
3	0	C	0
4	5	D	6
5	0	E	0
6	0	F	0
7	0	G	0

Σχήμα 10.5. (α) *Ένα δυαδικό δένδρο και* ***(β)*** *η αναπαράστασή του με πίνακα σύνθετου τύπου δεδομένων*

10.3.1 Έμμεση αναπαράσταση δυαδικού δένδρου σε πίνακα

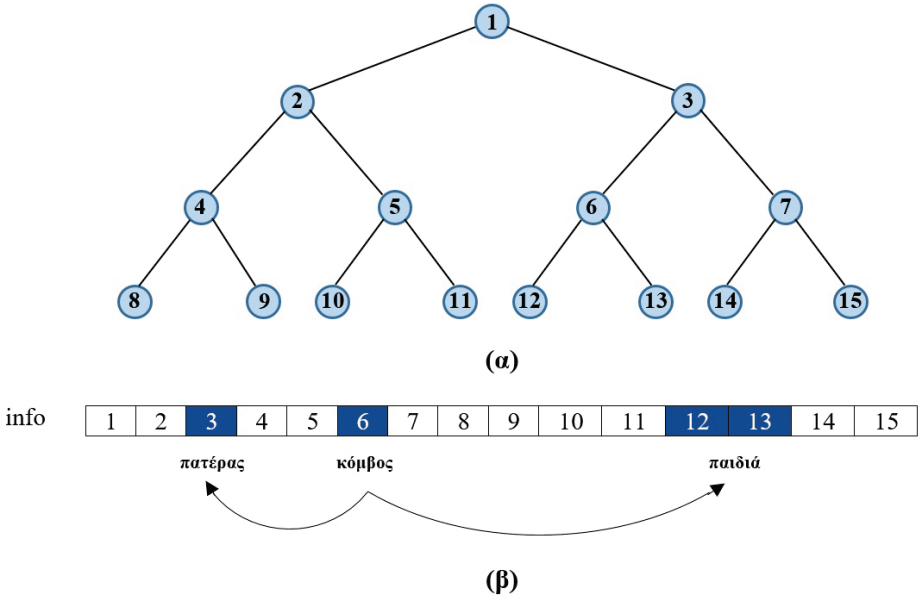
Ένας εναλλακτικός τρόπος αποθήκευσης σε πίνακα, που εφαρμόζεται στα δυαδικά δένδρα. κάνει χρήση ενός απλούστερου πίνακα σε σχέση με αυτόν που είδαμε στην προηγούμενη ενότητα. Ο πίνακας αυτός που περιέχει μόνο τα δεδομένα, ενώ οι ιεραρχικές συσχετίσεις μεταξύ κόμβων ***υπονοούνται*** από τις θέσεις τους στον πίνακα, για αυτό και αναφέρεται στη βιβλιογραφία ως ***έμμεση αναπαράσταση*** (*implicit representation*) ενός δυαδικού δένδρου σε πίνακα.

Η συγκεκριμένη αναπαράσταση στηρίζεται στην παρατήρηση ότι είναι πολύ αποδοτικό να κρατηθούν $2^{h+1} - 1$ θέσεις μνήμης, όπου **h** το ύψος του δένδρου, ιδιαίτερα αν το δένδρο προσεγγίζει το ***τέλειο*** (*perfect*) δένδρο. Σε κάθε θέση αποθηκεύονται τα δεδομένα ενός κόμβου και η σχέση με τον πατέρα και τα παιδιά του καθορίζεται από τις ίδιες τις θέσεις.

Η πολιτική αποθήκευσης του δυαδικού δένδρου στον πίνακα είναι απλή:

- Η ***ρίζα*** αποθηκεύεται στην πρώτη θέση του πίνακα (εδώ θα θεωρήσουμε ότι πρόκειται για τη θέση 1): $i = 1$.
- Το ***αριστερό παιδί*** κόμβου που είναι αποθηκευμένος στη θέση i του πίνακα, αν υπάρχει, θα είναι αποθηκευμένο στη θέση: $child_L(i) = 2 \cdot i$
- Το ***δεξί παιδί*** κόμβου που είναι αποθηκευμένος στη θέση i του πίνακα, αν υπάρχει, θα είναι αποθηκευμένο στη θέση: $child_R(i) = 2 \cdot i + 1$

Η απλή αυτή πολιτική οδηγεί στην αντιστοίχιση, με μονοσήμαντο τρόπο, οποιουδήποτε κόμβου του δένδρου σε μία μοναδική θέση του πίνακα. Εναλλακτικά, η παραπάνω πολιτική οδηγεί σε μία μοναδική αρίθμηση των κόμβων του δυαδικού δένδρου που στη συνέχεια αξιοποιείται για την επιλογή θέσης στον πίνακα.



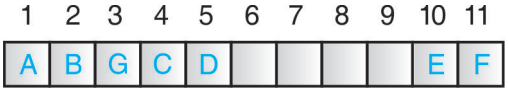
Σχήμα 10.6. (α) *Ένα τέλειο δυαδικό δένδρο και* ***(β)*** *η αναπαράστασή του σε έναν πίνακα.*

Συγκεκριμένα, αν ακολουθήσουμε την πολιτική αυτή, το αριστερό παιδί της ρίζας θα βρίσκεται στη θέση 2 ενώ το δεξί της παιδί στη θέση 3. Συνεχίζοντας, το αριστερό παιδί του κόμβου στη θέση 2 θα βρίσκεται στη θέση 4 ενώ το δεξί του παιδί στη θέση 5 κ.ο.κ. Στο σχήμα 10.6 φαίνεται η αρίθμηση στην οποία οδηγεί η παραπάνω πολιτική για τους κόμβους ενός τέλειου δυαδικού δένδρου ύψους 3 και η

αντίστοιχη αποθήκευσή τους στις θέσεις ενός πίνακα. Επίσης, στην πράξη, στην περίπτωση που η αρίθμηση υπερβαίνει τα όρια του πίνακα, υπονοείται ότι ο αντίστοιχος κόμβος δεν υπάρχει στο δένδρο

Ένα σημαντικό παρεπόμενο της παραπάνω πολιτικής είναι ότι μπορούμε άμεσα να εντοπίσουμε, εκτός από τα παιδιά, και τον γονέα οποιουδήποτε κόμβου. Συγκεκριμένα, για οποιονδήποτε κόμβο στη θέση (ή, αλλιώς, με αρίθμηση) ***i***, ο γονέας του βρίσκεται στη θέση (ή, αλλιώς, έχει αρίθμηση) $\lfloor i/2 \rfloor$. Για παράδειγμα, ο γονέας του κόμβου 13 είναι ο $\lfloor 13/2 \rfloor = \lfloor 6{,}5 \rfloor = 6$. Για $i = 1$, προφανώς πρόκειται για τη ρίζα για αυτό και οδηγούμαστε στην τιμή 0, αφού δεν υπάρχει γονέας. Όπως γίνεται αντιληπτό, η πολιτική μπορεί να αξιοποιηθεί και για εντοπισμό παραπέρα προγόνων και απογόνων με την κατάλληλη ακολουθία πράξεων σε σχέση με τους δείκτες του πίνακα.

Ο τρόπος αυτός μπορεί να χρησιμοποιηθεί για οποιοδήποτε δυαδικό δένδρο, χωρίς την απαίτηση να είναι τέλειο. Για παράδειγμα, στο σχήμα 10.7 φαίνεται η αποθήκευση του δένδρου του σχήματος 10.5α με τον τρόπο αυτό. Είναι φανερό ότι σε αυτήν την περίπτωση θα υπάρχει κάποια σπατάλη μνήμης που εξαρτάται από την κατανομή των κόμβων στο δένδρο αφού οι δείκτες αυξάνονται για να εξασφαλίσουν την τήρηση της συσχέτισης μεταξύ κόμβων, αφήνοντας ενδιάμεσα κενά. Από την άλλη μεριά όμως, απαιτείται ένας απλούστερος πίνακας, αφού δεν αποθηκεύονται δείκτες για τις θέσεις παιδιών. Πρακτικά, επαφίεται στο σχεδιαστή της εφαρμογής να αξιολογήσει τα πλεονεκτήματα και μειονεκτήματα κάθε προσέγγισης και να επιλέξει ποιον τρόπο θα ακολουθήσει.



Σχήμα 10.7. *Η αποθήκευση του δένδρου του σχήματος 10.5α με χρήση ενός μόνο πίνακα*

Κλείνοντας, αξίζει να σημειώσουμε ότι αντίστοιχοι τρόποι έμμεσης αποθήκευσης μπορούν να εφαρμοστούν σε δένδρα οποιουδήποτε βαθμού d. Όπως είναι αναμενόμενο, στις περιπτώσεις αυτές, οι παραστάσεις που υποδεικνύουν τη θέση κάθε παιδιού k από τα πιθανά d του εκάστοτε κόμβου στη θέση i, γενικεύονται, σε σχέση με αυτές που είδαμε για το δυαδικό δένδρο, ως εξής:

$$child_k(i) = d \cdot (i-1) + k + 1, \qquad 1 \le k \le d$$

Με βάση την παραπάνω γενίκευση, αντικαθιστώντας με $d = 2$ και $k = 1$ για αριστερό ή $k = 2$ για δεξί παιδί, προκύπτουν οι μορφές του δυαδικού δένδρου.

10.3.2 Συμπληρωμένο (complete) δυαδικό δένδρο

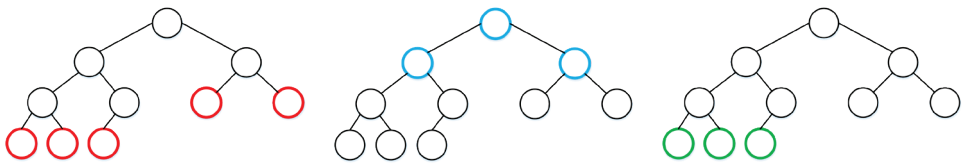
Αποτέλεσμα των παραπάνω προβλημάτων είναι ότι ο προηγούμενος τρόπος έμμεσης αναπαράστασης πιο συχνά συνδυάζεται και εξυπηρετεί δυαδικά δένδρα συγκεκριμένης μορφολογίας. Ένα τέτοιο δένδρο είναι το *συμπληρωμένο* (complete) δυαδικό δένδρο[4]. Το συγκεκριμένο είδος δένδρου υπακούει κάποιες θεωρητικές αρχές που, ωστόσο, έχουν έναν πολύ χρήσιμο πρακτικό αντίκτυπο. Ας δούμε πρώτα το θεωρητικό του ορισμό με βάση τις παρακάτω απαιτούμενες ιδιότητες.

Ιδιότητα 1. Ένα συμπληρωμένο (complete) δυαδικό δένδρο ύψους h έχει όλους τους τερματικούς του κόμβους (φύλλα) στα δύο χαμηλότερα επίπεδα, $h-1$ και h

Ιδιότητα 2. Όλοι οι κόμβοι από το επίπεδο 0 μέχρι και το επίπεδο $h-2$ έχουν ακριβώς δύο παιδιά

Ιδιότητα 3. Οι (τερματικοί) κόμβοι του h-στού (χαμηλότερου) επιπέδου καταλαμβάνουν τις αριστερότερες θέσεις αυτού του επιπέδου

Το παρακάτω σχήμα συνοψίζει τις ιδιότητες που απαιτεί ο ορισμός ενός συμπληρωμένου δένδρου και τους κόμβους που σχετίζονται με κάθε μία από αυτές.

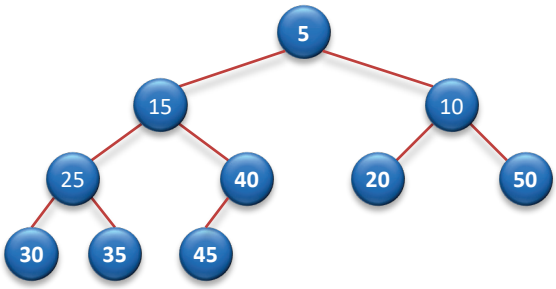


Ιδιότητα 1 *Ιδιότητα 2* *Ιδιότητα 3*

Σχήμα 10.8. *Παράδειγμα συμπληρωμένου (complete) δένδρου και κόμβοι που σχετίζονται με κάθε μία από τις ιδιότητές του*

Ο παραπάνω σχετικά περίπλοκος ορισμός εξασφαλίζει στα ***συμπληρωμένα*** (*complete*) δένδρα μία πολύ ενδιαφέρουσα και χρήσιμη ιδιότητα. Συγκεκριμένα, παρότι αυτό το είδος δένδρου δεν είναι τέλειο και η έμμεση αναπαράστασή του σε πίνακα αναμένεται να αφήσει κενά, οι προδιαγραφές του εξασφαλίζουν ότι τα οποιαδήποτε κενά είναι συγκεντρωμένα στο τέλος του πίνακα. Πράγματι, το δένδρο του προηγούμενου σχήματος, αποθηκευμένο σε πίνακα με έμμεση αναπαράσταση, εμφανίζεται στο επόμενο σχήμα:

[4] Στις περιπτώσεις της βιβλιογραφίας που ένα τέλειο δένδρο αποκαλείται πλήρες δένδρο, το συγκεκριμένο είδος αναφέρεται ως ***σχεδόν πλήρες*** (*almost complete*).



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
5	15	10	25	40	20	50	30	35	45					

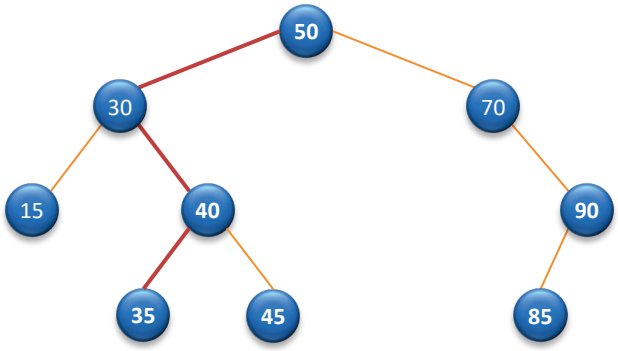
Σχήμα 10.9. *Παράδειγμα συμπληρωμένου (complete) δένδρου και έμμεση (implicit) αναπαράστασή του σε πίνακα*

Θα δούμε αργότερα αναλυτικά, παρουσιάζοντας και έναν αντιπροσωπευτικό αλγόριθμο που βασίζεται σε συμπληρωμένο δένδρο, ότι η συγκεκριμένη αναπαράσταση μπορεί να υποκαταστήσει με επιτυχία την όποια δυναμική εκδοχή που βασίζεται σε δείκτες, αξιοποιώντας την για μία νέα δενδρική δομή. Αυτό, ωστόσο, μπορεί να συμβεί υπό την αρκετά περιοριστική προϋπόθεση, ότι καταφέρνουμε τηρήσουμε τις παραπάνω ιδιότητες του δένδρου, σε όλη την εξέλιξη της ενημέρωσής του κατά την εκτέλεση του εκάστοτε αλγορίθμου και μάλιστα με αποδοτικό τρόπο.

10.4 Δυαδικά δένδρα αναζήτησης

Ίσως η πλέον ενδιαφέρουσα κατηγορία δένδρων είναι τα ***δυαδικά δένδρα αναζήτησης*** (*binary search trees – BSTs*). Τα δυαδικά δένδρα αναζήτησης είναι ***διατεταγμένα δένδρα***. Διατεταγμένο λέγεται το δένδρο στο οποίο έχει σημασία η διάταξη των παιδιών κάθε κόμβου. Σε επόμενο κεφάλαιο θα δούμε και διατεταγμένα δένδρα μεγαλύτερου βαθμού από τα δυαδικά δένδρα που θα εξετάσουμε εδώ.

Δυαδικό δένδρο αναζήτησης είναι το δένδρο, του οποίου η τιμή κάθε κόμβου είναι μεγαλύτερη από την τιμή όλων των κόμβων του αριστερού του υποδένδρου και μικρότερη από την τιμή όλων των κόμβων του δεξιού του υποδένδρου.



Σχήμα 10.10. *Ένα δυαδικό δένδρο αναζήτησης*

Αν ένα δυαδικό δένδρο αναζήτησης είναι επιθυμητό να περιέχει διπλότυπες τιμές ο παραπάνω κανόνας θα περιλαμβάνει την ισότητα σε μία από τις δύο κατευθύνσεις. Συνήθως, στην περίπτωση που είναι επιθυμητά τα ***διπλότυπα***, η τιμή κάθε κόμβου είναι ***μεγαλύτερη ή ίση*** από την τιμή όλων των κόμβων του ***αριστερού*** του υποδένδρου και ***μικρότερη*** από την τιμή όλων των κόμβων του ***δεξιού*** του υπο-δένδρου. Ωστόσο, είναι σημαντικό να τονίσουμε ότι η όποια επιλογή για την κατεύθυνση της ισότητας (εδώ προς το αριστερό υποδένδρο) θα πρέπει να τηρείται με αυστηρή συνέπεια και να είναι κοινή σε όλους τους κόμβους του δένδρου. Με άλλα λόγια, η ισότητα δεν μπορεί να εναλλάσσεται μεταξύ αριστερού και δεξιού υποδένδρου, για διαφορετικούς κόμβους του ίδιου δένδρου.

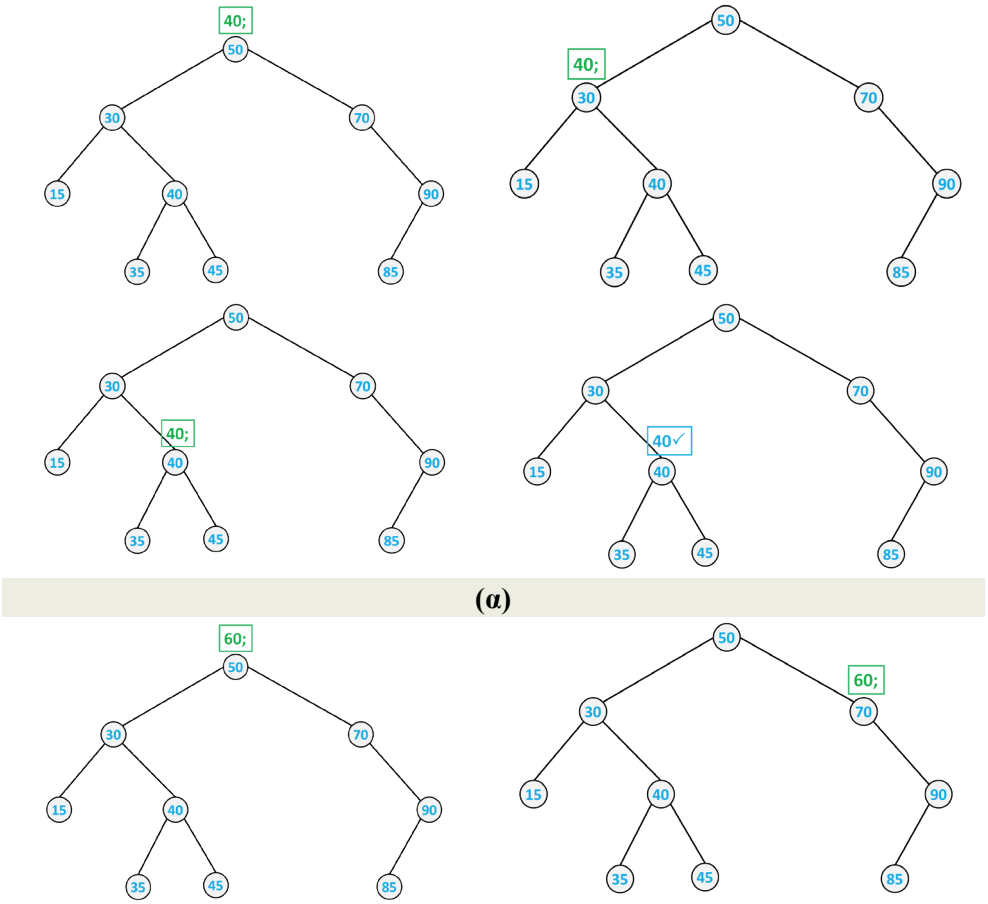
Τα δυαδικά δένδρα αναζήτησης χρησιμοποιούνται όταν είναι σημαντική η γρήγορη προσπέλαση δεδομένων, που υπακούει σε κάποια σειρά κατάταξης. Διαισθητικά μπορούμε να θεωρήσουμε ότι τα δυαδικά δένδρα αναζήτησης αποτελούν μία πρακτική υλοποίηση, σε επίπεδο δομής δεδομένων, της φιλοσοφίας του αλγόριθμου δυαδικής αναζήτησης. Ωστόσο, ενώ ο αλγόριθμος δυαδικής αναζήτησης καταφέρνει σε κάθε βήμα να μειώσει το μέγεθος του προβλήματος στο μισό, ένα τυχαίο δυαδικό δένδρο αναζήτησης είναι πιθανό να μην το καταφέρνει αυτό χωρίς κάποια επιπλέον επεξεργασία, λόγω του τρόπου που εισάγονται σε αυτό τα δεδομένα – θα δούμε αυτό το σημείο αναλυτικότερα στη συνέχεια του κεφαλαίου.

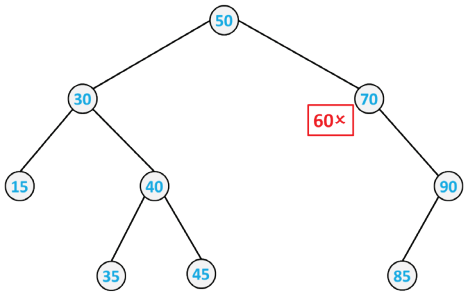
Ας σημειωθεί ότι στους κόμβους ενός δένδρου μπορεί να αντιστοιχούν εγγραφές που περιέχουν πληροφορίες, όπως η επωνυμία και η διεύθυνση πελατών, η περιγραφή, η τιμή μονάδος και η ποσότητα ειδών κ.λπ. Η τιμή που εμφανίζεται στους κόμβους του δένδρου, μπορεί να θεωρηθεί ότι αντιστοιχεί στον κωδικό του πελάτη ή του είδους. Εντοπίζοντας έναν κόμβο με κάποιον κωδικό, γίνονται προσβάσιμες και οι υπόλοιπες πληροφορίες. Ο κωδικός αυτός αποκαλείται ***κλειδί αναζήτησης***.

10.4.1 Αναζήτηση κόμβου

Για την εύρεση ενός συγκεκριμένου κόμβου σε ένα δυαδικό δένδρο αναζήτησης πρώτα εξετάζεται η τιμή της ρίζας του δένδρου σε σχέση με την τιμή του κόμβου που ζητάμε. Αν ο ζητούμενος κόμβος είναι μικρότερος από τη ρίζα συνεχίζουμε με το αριστερό υποδένδρο, ενώ αν είναι μεγαλύτερος, με το δεξί (φυσικά αν είναι ίσος, τότε η αναζήτηση τελειώνει). Η αναζήτηση σταματά, αν βρεθεί ο ζητούμενος κόμβος ή φτάσουμε σε κενό δείκτη.

Το παρακάτω σχήμα αναπαριστά οπτικά τα διαδοχικά βήματα μίας επιτυχημένης και μίας αποτυχημένης αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης.





(β)

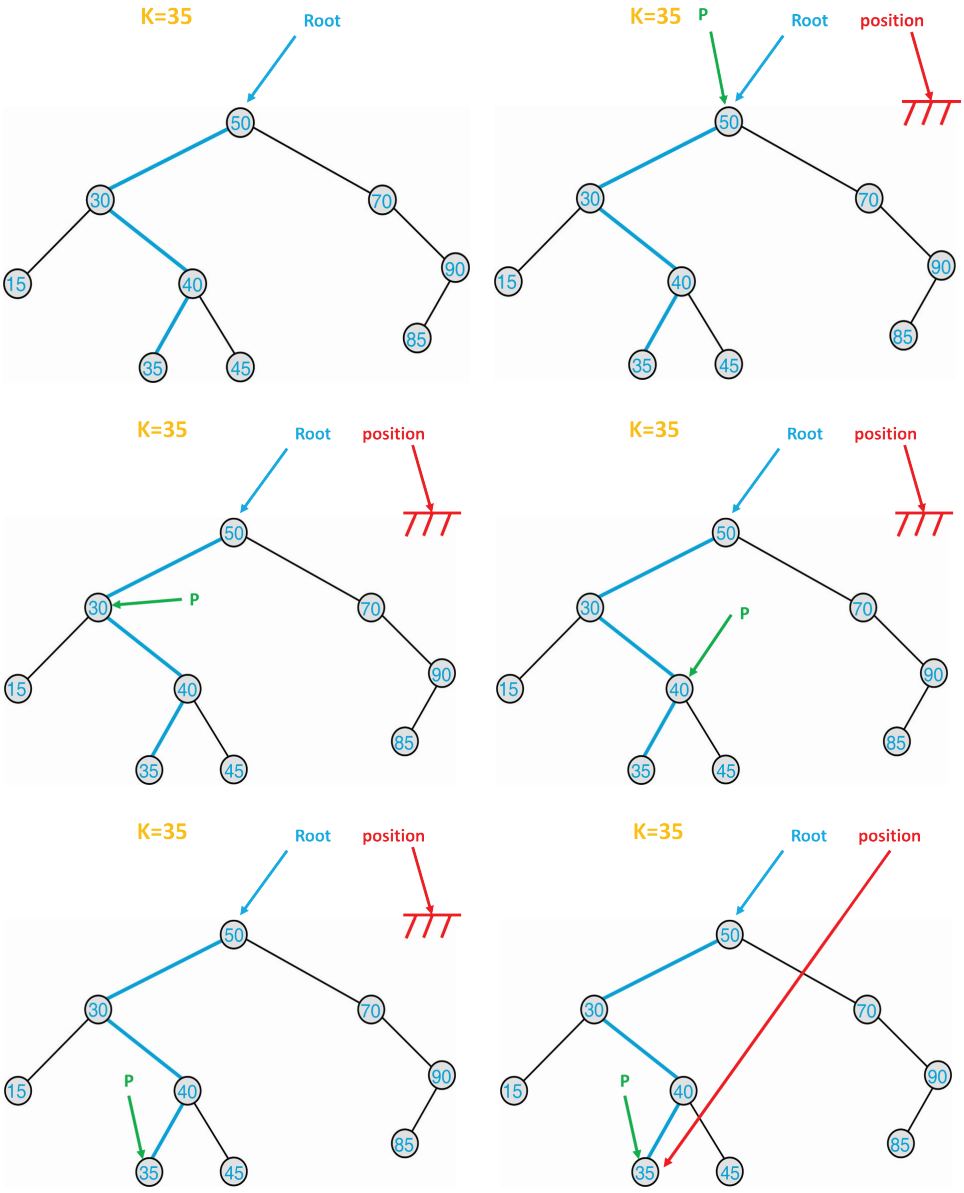
Σχήμα 10.11. *Οπτική αναπαράσταση* ***(α)*** *επιτυχημένης και* ***(β)*** *αποτυχημένης αναζήτησης σε δυαδικό δένδρο αναζήτησης*

Η υλοποίηση της αναζήτησης παρουσιάζεται στον αλγόριθμο Search-Node. Δεδομένα του αλγορίθμου αυτού είναι ο δείκτης ***Root***, που δείχνει τη ρίζα του δένδρου και η μεταβλητή ***K***, της οποίας η τιμή αναζητείται ως περιεχόμενο κάποιου κόμβου. Η αναπαράσταση του δένδρου ακολουθεί τη δυναμική εκδοχή, όπου κάθε κόμβος περιέχει τα δεδομένα με τη μορφή του πεδίου ***Info*** και δύο δείκτες στο αριστερό και δεξί παιδί αντίστοιχα με τη μορφή των πεδίων ***Left*** και ***Right***.

```
Αλγόριθμος Search-Node
Δεδομένα // Root, K //
P ← Root : found ← ψευδής : position ← nil
Όσο P ≠ nil και found = ψευδής επανάλαβε
        Αν K = P.Info τότε
                found ← αληθής : position ← P
        αλλιώς_αν K > P.Info τότε
                P ← P.Right
        αλλιώς
                P ← P.Left
        Τέλος_αν
Τέλος επανάληψης
Αποτελέσματα // found, position //
Τέλος Search-Node
```

Αναζήτηση κόμβου

Τα αποτελέσματα της αναζήτησης είναι και εδώ η λογική μεταβλητή ***found***, που σημειώνει αν η αναζήτηση υπήρξε επιτυχής ή όχι, και η ***position***, που παρέχει τη θέση του κόμβου, εφόσον βρέθηκε. Παρακάτω παρουσιάζεται σχηματικά η εξέλιξη των δεικτών σε κάθε βήμα του αλγορίθμου.



Σχήμα 10.12. *Εξέλιξη των δεικτών κατά την εκτέλεση του ψευδοκώδικα αναζήτησης (επιτυχημένη αναζήτηση)*

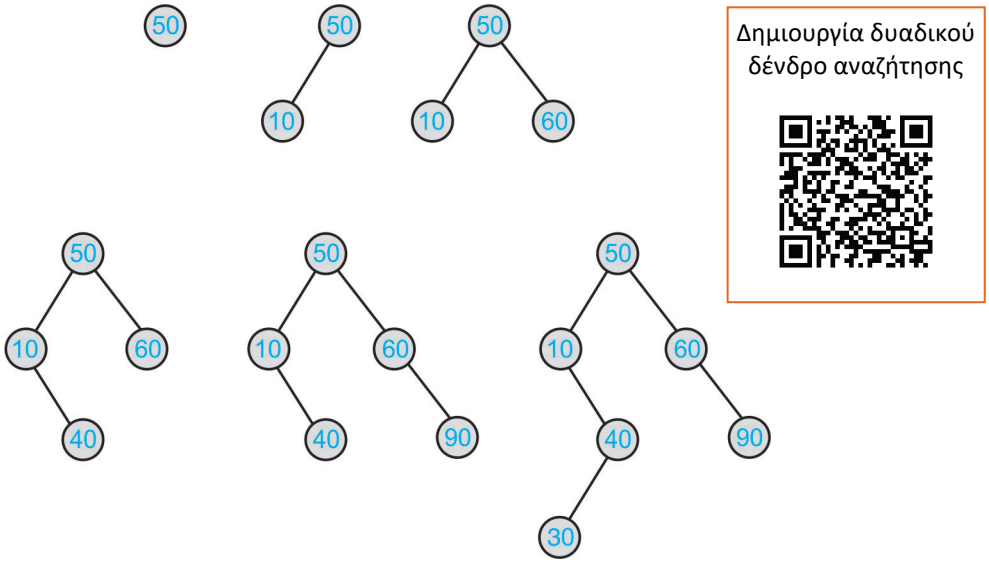
10.4.2 Εισαγωγή ή δημιουργία κόμβου

Σε ένα δυαδικό δένδρο αναζήτησης οι νέοι κόμβοι εντάσσονται *πάντα* ως τερματικοί κόμβοι. Η εισαγωγή ή δημιουργία κόμβου σε δυαδικό δένδρο αναζήτησης γίνεται με τον εξής τρόπο: Αρχικά ακολουθείται η διαδικασία της αναζήτησης, αναζητώντας τον κόμβο που πρέπει να προστεθεί. Ο κόμβος φυσικά δεν θα υπάρχει και η αναζήτηση οδηγείται σε έναν κενό δείκτη. Ο κόμβος προστίθεται με τρόπο ώστε ο κενός δείκτης τώρα να δείχνει σ' αυτόν.

Στην περίπτωση που θεωρούμε ότι δεχόμαστε τη δημιουργία διπλοτύπων στο δένδρο, τότε η αναζήτηση δεν ελέγχει την ισότητα ως ξεχωριστή περίπτωση: η ισότητα έχει ανατεθεί σε μία από τις δύο κατευθύνσεις (συνήθως προς τα αριστερά) και η τιμή που πρόκειται να προστεθεί στο δένδρο καθοδηγεί την επιλογή κατεύθυνσης μέχρι να βρεθεί κενός δείκτης. Ο νέος κόμβος προστίθεται με το ίδιο σκεπτικό: ο κενός δείκτης δείχνει πια σε αυτόν.

Στο σχήμα 10.10 φαίνεται έντονα σημειωμένη η διαδρομή αναζήτησης του κόμβου **35**. Αν έπρεπε να ενταχθεί ένας κόμβος με τιμή 33 ή 37, τότε θα έπρεπε να δημιουργηθεί ως αριστερό ή δεξιό παιδί, αντίστοιχα, του κόμβου 35.

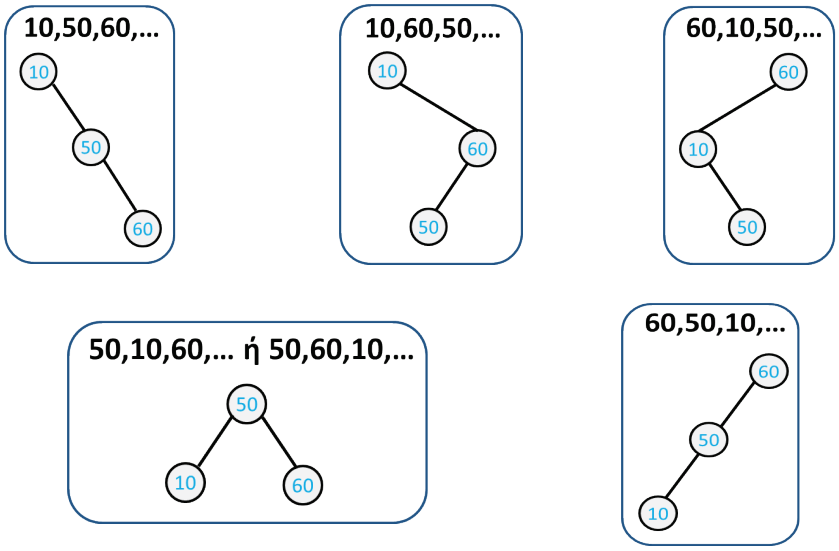
Για τη διαδικασία αναζήτησης μπορεί να χρησιμοποιηθεί ο Search_Node, αφού τροποποιηθεί κατάλληλα, ώστε να παρέχει ως αποτέλεσμα και τη θέση του πατέρα του προς δημιουργία κόμβου.



Σχήμα 10.13. *Αρχική δημιουργία ενός δένδρου αναζήτησης με τη διαδοχική εισαγωγή των κόμβων με τιμές 50, 10, 60, 40, 90, 30 (με τη σειρά αυτή).*

Όπως γίνεται φανερό και από το παράδειγμα του προηγούμενου σχήματος, η σειρά εισαγωγής των τιμών σε ένα δυαδικό δένδρο αναζήτησης επηρεάζει άμεσα τη μορφή που προκύπτει για αυτό. Μία πρώτη παρατήρηση που μπορούμε να κάνουμε είναι ότι αριθμητική απόσταση μεταξύ δύο τιμών δεν αποτελεί κάποιου είδους ένδειξη σχετικά με την απόστασή τους ως προς τα επίπεδα του δένδρου. Για παράδειγμα, η εισαγωγή της τιμής 71 στο δένδρο του σχήματος 10.10 θα την οδηγούσε να τοποθετηθεί ως αριστερό παιδί του κόμβου 85, αρκετά πιο κάτω από την τιμή 70, αφού αυτό υπαγορεύει η τρέχουσα κατάσταση του δένδρου.

Επεκτείνοντας αυτήν τη σκέψη, μπορούμε να παρατηρήσουμε ότι οι ίδιες ακριβώς τιμές, αλλά με διαφορετικές σειρές άφιξης, δημιουργούν διαφορετικά δυαδικά δένδρα αναζήτησης. Είναι χαρακτηριστικό ότι έστω και τρεις τιμές κλειδιών μπορούν να οδηγήσουν σε πέντε (!) εναλλακτικά πιθανά δυαδικά δένδρα αναζήτησης ανάλογα με τη σειρά άφιξής τους, όπως γίνεται φανερό από το παρακάτω σχήμα.



Σχήμα 10.14. *Εναλλακτικά δυαδικά δένδρα αναζήτησης που δημιουργούνται ανάλογα με τη σειρά άφιξης των τιμών 10, 50, 60*

Στα δεδομένα του αλγόριθμου δημιουργίας κόμβου, εκτός από το περιεχόμενο του νέου κόμβου, πρέπει να υπάρχει και η ***θέση μνήμης***, στην οποία θα καταχωρηθεί ο νέος κόμβος. Στην περίπτωση που ακολουθείται η υλοποίηση του δένδρου μέσω διασυνδεδεμένων κόμβων, στο θέμα αυτό ισχύουν όσα αναφέρθηκαν στην περίπτωση της δημιουργίας κόμβου λίστας (βλ. κεφ. 9) και τα οποία στις εφαρμογές ποικίλουν ανάλογα και με τον ακριβή τρόπο αποθήκευσης του δένδρου.

Όταν ένα δυαδικό δένδρο αναζήτησης πρόκειται να δημιουργηθεί για πρώτη φορά, τότε ο ***πρώτος κόμβος*** που προσέρχεται γίνεται οπωσδήποτε η ***ρίζα του δένδρου***. Ο επόμενος κόμβος γίνεται αριστερό ή δεξί παιδί της ρίζας ανάλογα με την τιμή του. Για κάθε επόμενο κόμβο ακολουθείται η προηγούμενη διαδικασία και εισάγεται πάντα ως τερματικός κόμβος αφού προστίθεται μετά τον εντοπισμό του κατάλληλου κενού δείκτη.

10.4.3 Χρήσιμες Ιδιότητες

Ο ορισμός και η κατασκευή ενός δυαδικού δένδρου αναζήτησης οδηγεί σε κάποιες ιδιότητες που ήδη αξιοποιήσαμε στις διαδικασίες αναζήτησης και εισαγωγής αλλά θα μας είναι χρήσιμες και στη συνέχεια, για τη λειτουργία της διαγραφής. Χωρίς βλάβη της γενικότητας, στη συνέχεια της ενότητας θα υποθέσουμε ότι αναφερόμαστε σε δυαδικά δένδρα αναζήτησης χωρίς διπλότυπα και ότι οι αναφορές σε τιμές κόμβων αφορούν τις τιμές κλειδιών που αυτοί περιέχουν και που καθοδηγούν την κατασκευή του δένδρου και τη συνακόλουθη αναζήτηση και εισαγωγή νέων τιμών σε αυτό. Με άλλα λόγια, οι τιμές στις οποίες αναφερόμαστε είναι διατεταγμένες – ένα τέτοιο δυαδικό δένδρο αναζήτησης, πρακτικά, υπονοεί την ολική διάταξη των περιεχομένων του.

Πιο συγκεκριμένα, αξιοποιήθηκε ήδη στις προηγούμενες λειτουργίες εισαγωγής και αναζήτησης το γεγονός ότι, για οποιοδήποτε κόμβο, οι κόμβοι που ανήκουν στο αριστερό του υποδένδρο, δηλαδή στο υποδένδρο με ρίζα το αριστερό του παιδί, αν αυτό υπάρχει, έχουν όλοι μικρότερες τιμές. Συμμετρικά, για οποιοδήποτε κόμβο, οι κόμβοι που ανήκουν στο δεξί του υποδένδρο, αν αυτό υπάρχει, έχουν όλοι μεγαλύτερες τιμές.

Βασισμένοι στην παραπάνω παρατήρηση μπορούμε να εντοπίσουμε σχετικά εύκολα τον κόμβο που περιέχει την ελάχιστη τιμή του δένδρου: θα είναι ο «αριστερότερος» κόμβος του δένδρου. Συγκεκριμένα, ξεκινώντας από τη ρίζα και επιλέγοντας διαδοχικά το αριστερό παιδί του εκάστοτε τρέχοντα κόμβου, εξασφαλίζουμε ότι οδηγούμαστε στον κόμβο με την ελάχιστη τιμή. Πράγματι, όταν, κατά τη διαδικασία αυτή, οδηγηθούμε σε έναν κόμβο χωρίς αριστερό παιδί, τότε είμαστε βέβαιοι ότι βρισκόμαστε στον κόμβο με την ελάχιστη τιμή. Αυτό ισχύει γιατί, αν τυχόν υπήρχε κόμβος με μικρότερη τιμή από αυτή του τρέχοντα κόμβου τότε αυτή (α) είτε θα έπρεπε να περιέχεται στο αριστερό υποδένδρο του τρέχοντα κόμβου, που ήδη υποθέσαμε ότι δεν υπάρχει, για αυτό και σταματήσαμε στον τρέχοντα κόμβο, (β) είτε θα έπρεπε να περιέχεται στο αριστερό υποδένδρο κάποιου προγόνου του τρέχοντα κόμβου, στο οποίο θα έπρεπε να μην ανήκει ο τρέχων κόμβος, κάτι που πάλι δεν είναι δυνατό να ισχύει αφού βρεθήκαμε στον τρέχοντα κόμβο επιλέγοντας πάντα το

αριστερό υποδένδρο του εκάστοτε προγόνου, με άλλα λόγια ο τρέχων κόμβος ***ανήκει στα αριστερά υποδένδρα όλων των προγόνων του***.

Η μέθοδος αυτή μπορεί βέβαια να αξιοποιηθεί συμμετρικά και για τον εντοπισμό της μέγιστης τιμής στο δυαδικό δένδρο αναζήτησης, επιλέγοντας τον «***δεξιότερο***» κόμβο του δένδρου.

Είναι, ωστόσο, ενδιαφέρον να αξιοποιήσουμε την ιδιότητα αυτή για τον εντοπισμό της προηγούμενης ή της επόμενης τιμής (ως προς τη διάταξη των τιμών του δένδρου) οποιουδήποτε κόμβου του δένδρου. Πριν προχωρήσουμε σε αυτήν τη συζήτηση αξίζει να κάνουμε μία απλή παρατήρηση: σε μία διατεταγμένη σειρά αριθμών, ο αμέσως προηγούμενος οποιουδήποτε αριθμού της σειράς, αν υπάρχει, είναι και ο ***μέγιστος*** από όλους τους αριθμούς που είναι μικρότεροι από αυτόν. Η απλή αυτή παρατήρηση θα μας καθοδηγήσει για τον καθορισμό της μεθόδου για τον εντοπισμό της προηγούμενης τιμής οποιουδήποτε κόμβου σε ένα δυαδικό δένδρο αναζήτησης. Κατόπιν, εντελώς συμμετρικά, θα μπορέσουμε να καθορίσουμε και τη μέθοδο για τον εντοπισμό της επόμενης τιμής.

Ας επικεντρωθούμε πρώτα στην απλούστερη περίπτωση που ο κόμβος διαθέτει αριστερό παιδί. Στην περίπτωση αυτή, η προηγούμενη τιμή θα ανήκει στο αριστερό του υποδένδρο. Μάλιστα, βασισμένοι στην παραπάνω απλή παρατήρηση για τη διάταξη αριθμών, η προηγούμενη αριθμητική τιμή του κόμβου θα είναι η ***μέγιστη*** τιμή στο ***αριστερό*** του ***υποδένδρο***. Όμως, σύμφωνα με την προηγούμενη συζήτηση περί μέγιστου, η μέγιστη αυτή τιμή θα είναι η «***δεξιότερη***» τιμή στο αριστερό του υποδένδρο, συνεπώς μπορούμε εύκολα να την εντοπίσουμε. Εντελώς συμμετρικά, η επόμενη τιμή ενός κόμβου που διαθέτει δεξί παιδί είναι η ελάχιστη τιμή, δηλαδή η «***αριστερότερη***», στο δεξί του υποδένδρο.

Επιστρέφοντας στην αναζήτηση της προηγούμενης τιμής, τα πράγματα γίνονται πιο περίπλοκα όταν ο κόμβος δεν διαθέτει αριστερό παιδί. Αν πρόκειται για τη ρίζα, δεν χρειάζεται καμία διερεύνηση αφού μία ρίζα χωρίς αριστερό παιδί υπονοείται ότι περιέχει την ελάχιστη τιμή του δυαδικού δένδρου αναζήτησης, αφού δεν μπορούμε να κινηθούμε προς τα αριστερά.

Ωστόσο, στην περίπτωση που ο κόμβος δεν διαθέτει αριστερό παιδί αλλά δεν είναι και η ρίζα του δένδρου, θα πρέπει να εντοπίσουμε έναν κόμβο που δεν ανήκει σε υποδένδρο του τρέχοντα κόμβου, αφού αυτό δεν υπάρχει, για να εντοπίσουμε τον προηγούμενο κόμβο. Προχωρώντας λίγο τη σκέψη μας με βάση τα όσα αναλύσαμε αμέσως πριν, ο κόμβος με την ***προηγούμενη*** τιμή θα είναι εκείνος για τον οποίο ο τρέχων κόμβος είναι ο ***επόμενος***. Αυτό συμβαίνει για τον πλησιέστερο πρόγονο του τρέχοντα κόμβου που έχει δεξί παιδί (και επομένως ο τρέχων κόμβος ανήκει στο