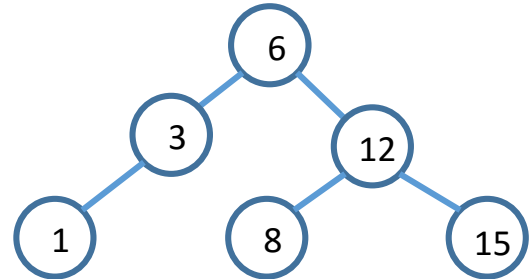


Δεύτερη Σειρά ασκήσεων Διαδικό Δέντρο Αναζήτησης – Binary Search Tree

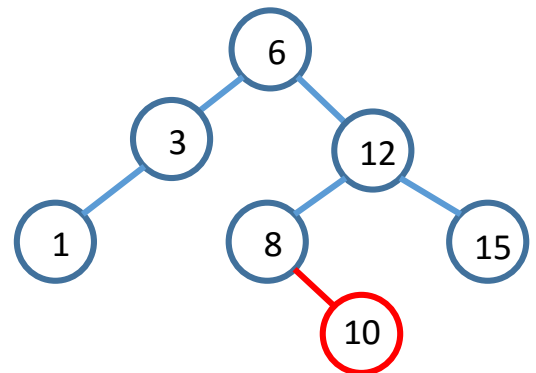
Το δυαδικό δέντρο είναι μια δομή που αποτελείται από **κόμβους** που συνδέονται μεταξύ τους. Ο κάθε κόμβος έχει μέχρι και δύο παιδιά και ένα γονέα. Ένας κόμβος χωρίς παιδιά λέγεται **φύλο**, ενώ υπάρχει και ένας κόμβος που είναι η ρίζα του **δέντρου** ο οποίος δεν έχει γονέα. Στο παράδειγμα του δυαδικού δέντρου που φαίνεται παρακάτω ο κόμβος με την τιμή 6 είναι η ρίζα, ενώ οι 1, 8, 15 είναι φύλα. Ο 3 και ο 12 είναι παιδιά του 6: το 3 είναι το **αριστερό παιδί**, και το 12 το **δεξί παιδί**. Ο 6 είναι ο γονέας τους. Για κάθε κόμβο στο δέντρο μπορούμε να ορίσουμε ένα δέντρο με ρίζα αυτόν τον κόμβο, και όλους τους κόμβους που είναι κάτω από αυτόν τον κόμβο. Αυτό το δέντρο το λέμε το **υποδέντρο** του κόμβου.



Στην υλοποίησή σας οι κόμβοι θα κρατάνε ακέραιες τιμές, όπως φαίνεται και στο παράδειγμα. Οι τιμές έχουν την ιδιότητα ότι για κάθε κόμβο όλες οι τιμές που είναι στο υποδέντρο αριστερά του κόμβου έχουν τιμές μικρότερες από την τιμή του κόμβου, ενώ όλες οι τιμές που είναι στο υποδέντρο δεξιά του κόμβου, έχουν τιμές μεγαλύτερες από αυτές του κόμβου.

Η **αναζήτηση** μίας τιμής x στο δέντρο ξεκινάει από την ρίζα. Αν το x είναι ίδιο με την τιμή της ρίζας έχουμε βρει την τιμή. Αλλιώς ανάλογα με το αν το x είναι μικρότερο ή μεγαλύτερο συνεχίζουμε την αναζήτηση στο αριστερό ή στο δεξιό υποδέντρο αντίστοιχα. Αν σταματήσουμε σε κόμβο που δεν έχει την τιμή τότε η τιμή δεν περιέχεται στο δέντρο.

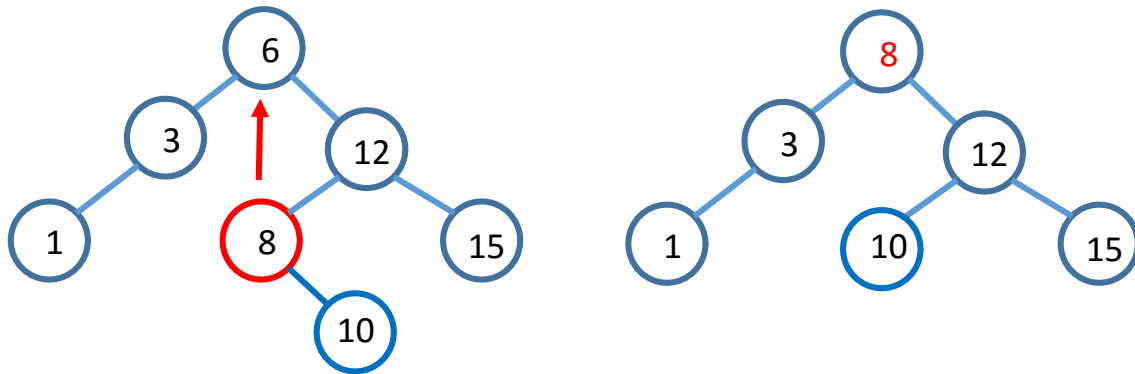
Η **εισαγωγή** μια νέας τιμής x στο δέντρο γίνεται προσθέτοντας ένα επιπλέον κόμβο ο οποίος θα κρατάει αυτή την τιμή. Ο κόμβος αυτός προστίθεται **πάντα** σαν φύλο. Η θέση του νέου φύλου γίνεται κάνοντας την ίδια διαδικασία όπως και στην αναζήτηση. Αν βρούμε την τιμή σε κάποιο κόμβο τότε σταματάμε την διαδικασία και δεν προσθέτουμε την τιμή. Αν σταματήσει η αναζήτηση χωρίς να βρούμε την τιμή, προσθέτουμε το νέο ως παιδί του κόμβου στον οποίο σταμάτησε η αναζήτηση. Στο διπλανό παράδειγμα φαίνεται το αποτέλεσμα της εισαγωγής της τιμής 10 στο προηγούμενο δέντρο.



Η **εξαγωγή** μιας τιμής x είναι πιο σύνθετη και χρειάζεται επιμέρους περιπτώσεις.

- Αν η τιμή που θέλουμε να αφαιρέσουμε βρίσκεται σε **φύλο** (π.χ., η τιμή 1) τότε απλά αφαιρούμε τον κόμβο από το δέντρο (τον αποκόπτουμε από τον γονέα του).
- Αν η τιμή που θέλουμε να αφαιρέσουμε ανήκει σε κόμβο με ένα μόνο παιδί (π.χ., η τιμή 3), τότε απλά παρακάμπτουμε τον κόμβο αυτό και βάζουμε τον πατέρα του να δείχνει στο παιδί του.
- Αν η τιμή που θέλουμε να αφαιρέσουμε ανήκει σε κόμβο με δύο παιδιά (π.χ., την τιμή 6), τότε βρίσκουμε τον κόμβο στο δεξιό υποδέντρο που έχει την **μικρότερη** τιμή, αντιγράφουμε τη τιμή αυτή, και μετά διαγράφουμε αυτό τον κόμβο. Για να βρούμε την μικρότερη τιμή μέσα στο δεξιό υποδέντρο, ξεκινάμε από τη ρίζα (στην περίπτωση μας, τον κόμβο 12) και πηγαίνουμε προς τα αριστερά όσο πιο χαμηλά γίνεται (στην περίπτωση αυτή θα σταματήσουμε στον κόμβο 8). Ο κόμβος αυτός υποχρεωτικά είτε θα είναι φύλο, είτε θα έχει ένα μόνο παιδί,

οπότε μπορούμε να τον αφαιρέσουμε όπως είδαμε πιο πάνω. Σχηματικά η διαγραφή της τιμής 6 περιγράφεται στο παρακάτω σχήμα.



Στην υλοποίηση σας δώσετε προσοχή στην ενημέρωση και του πεδίου του γονέα σε κάθε κόμβο, καθώς και της ειδικής περίπτωσης που αφαιρείτε την τιμή της ρίζας. Όταν ένας κόμβος δεν έχει κάποιο παιδί (ή στην περίπτωση της ρίζας δεν έχει γονέα) η τιμή του αντίστοιχου πεδίου θα είναι null.