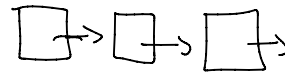


ουρές



συναρτήσεις

Διαδικασία Δέντρο Αναζήτησης
Συμφοί

ID	Name	Surname	Age
12	Constantina	Kyriakou	19
23	Nikos	Sotiropoulos	25
34	Peter	Michaels	34
45	Mary	Jane	42
52	Foye	Oluokun	29

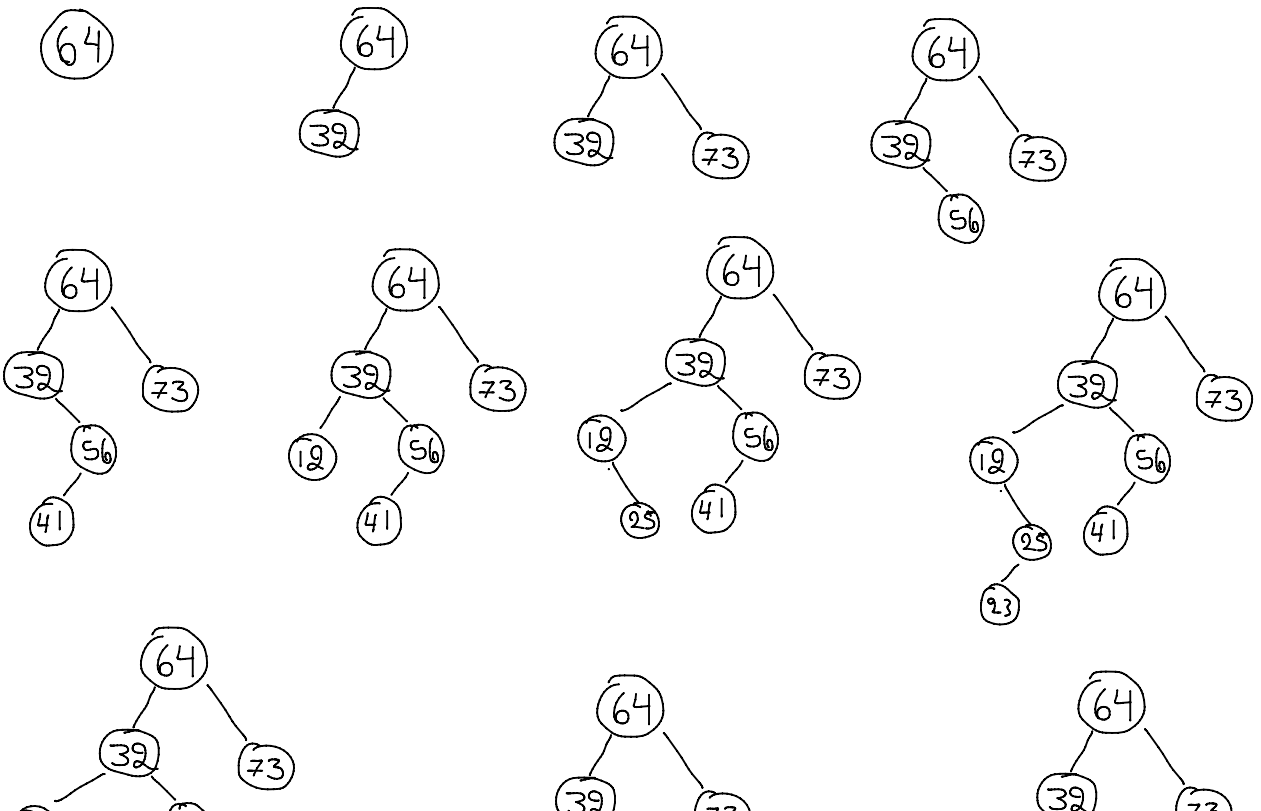
Αν έχω δεδομένα τα οποία μπορούν να ταξινομηθούν, τότε μπορώ να τα βάλω σε ένα δυαδικό δέντρο αναζήτησης (Binary Search Tree).

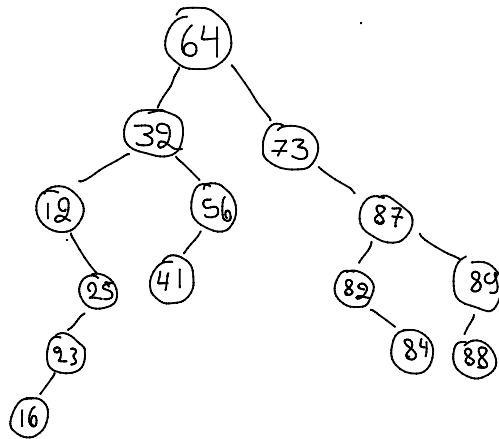
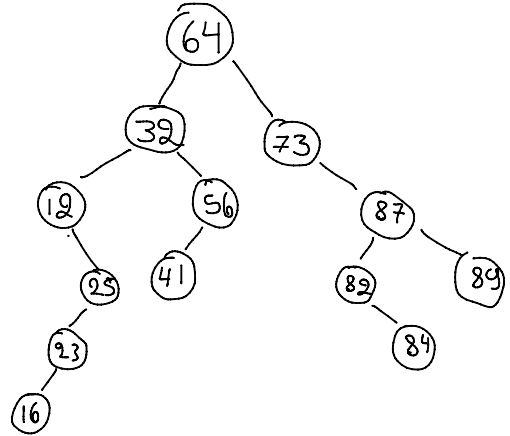
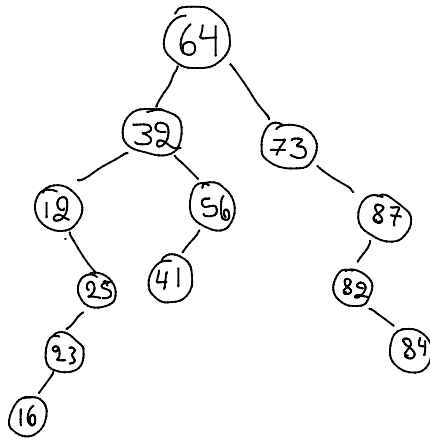
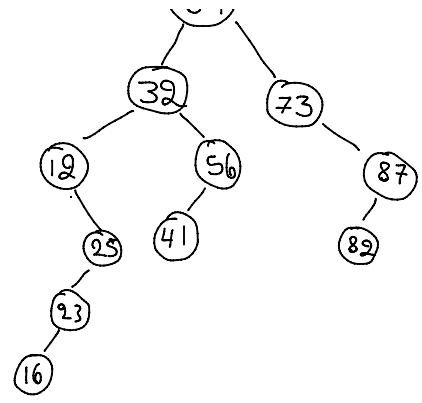
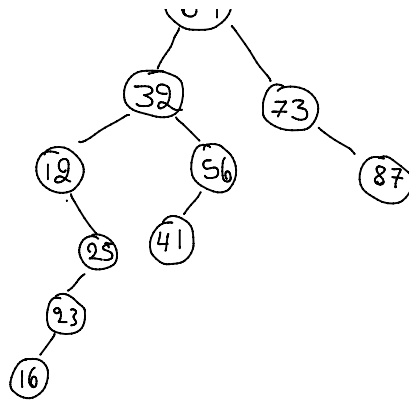
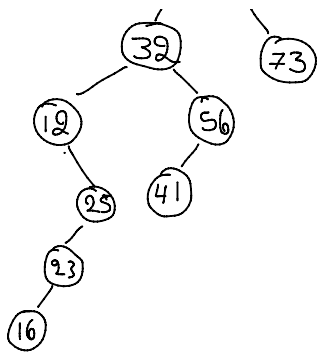
Στο δυαδικό δέντρο αναζήτησης κάθε κόμβος μπορεί να έχει το πολύ έναν γονιό και το πολύ δύο παιδιά. Το αριστερό παιδί ενός κόμβου θα πρέπει να έχει μικρότερη τιμή από τον κόμβο αυτό ενώ το δεξιό παιδί ενός κόμβου θα πρέπει να έχει μεγαλύτερη τιμή (ή και ίσες).

Πώς εισάγουμε στοιχεία στο δυαδικό δέντρο αναζήτησης;

64 32 73 56 41 12 25 23 16 87 82 84 89 88

Για το πρώτο στοιχείο το βάζουμε απλά στη ρίζα.





Αναζήτηση

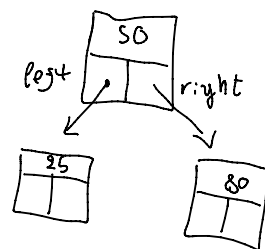
Δώσε μου το μονοπάτι που θα ακολουθήσω για να βρω το 82

Διασχίσεις

Ενδοδιατεταγμένη ΑΠΔ
Προδιατεταγμένη ΠΑΔ
Μεταδιατεταγμένη ΑΔΠ

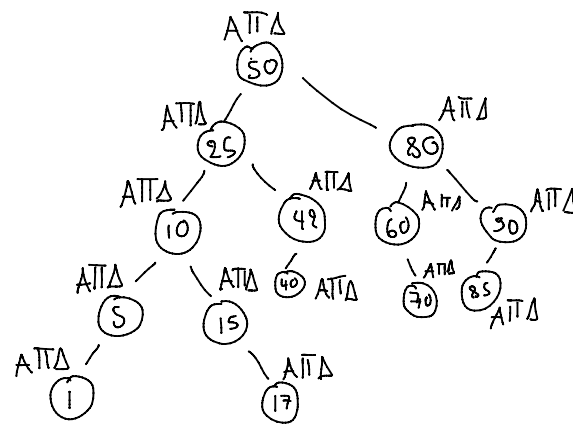
inorder

ΑΠΔ
(50)

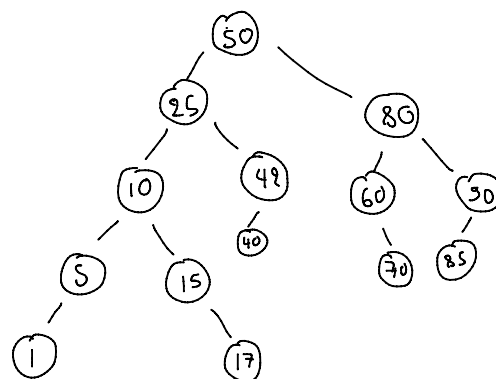


```
typedef struct node {
    int data;
    struct node *left;
    struct node *right;
} BSTNode;
```

1, 5, 10, 15, 17
25, 40, 42, 50, 60,



1, 5, 10, 15, 17
25, 40, 42, 50, 60,
70, 80, 85, 90

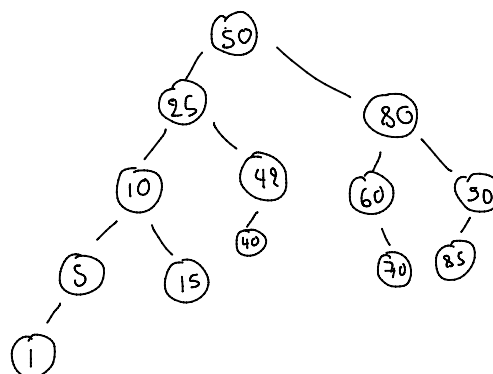


Predecessor ενός κόμβου A
Είναι ο μεγαλύτερος κόμβος
Του αριστερού υποδένδρου

Πχ predecessor(50) == 42

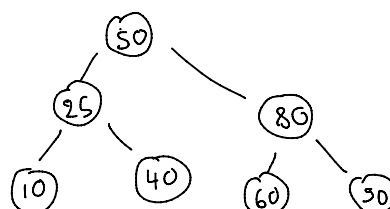
Διαγραφή

Αν θέλω να διαγράψω ένα φύλλο (κόμβος που δεν έχει παιδιά) τότε απλά το κάνω free και κάνω NULL τον pointer του πατέρα που δείχνει στο φύλλο

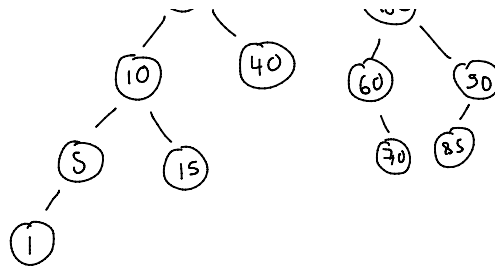


Πχ Διαγραφή του 17

Αν θέλω να διαγράψω έναν κόμβο που έχει μόνο ένα παιδί, ανεβάζω το παιδί στη θέση του πατέρα



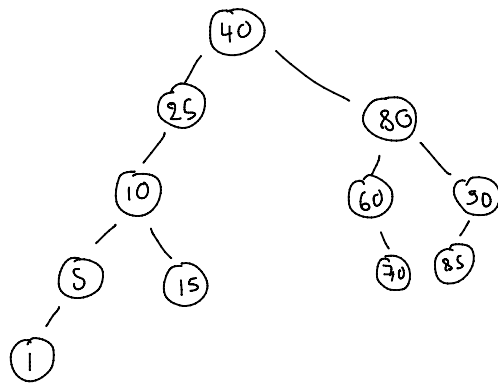
Πχ Διαγραφή του 42



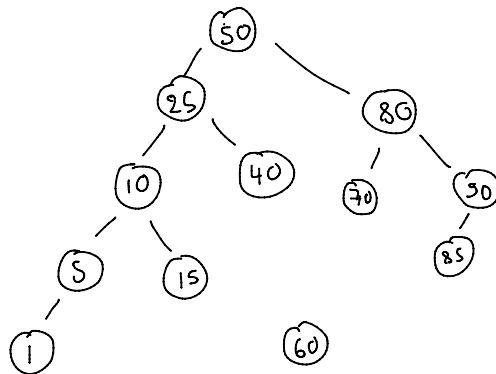
Αν θέλω να διαγράψω έναν κόμβο που έχει δύο παιδιά, βρίσκω είτε το successor του είτε τον predecessor του και τα ανεβάζω στη θέση του κόμβου αυτού.

Πχ Αν θέλω να διαγράψω το 50, μπορώ να βάλω είτε το 40 (predecessor) είτε το 60 (successor) στη θέση του 50.

ΠΑΡΑΤΗΡΗΣΗ: Ο successor και ο predecessor είναι πάντα φύλλο ή πατέρας με ένα παιδί!

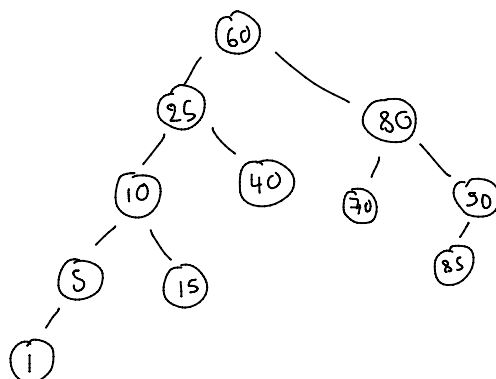


Πχ Διαγραφή του 50 με predecessor



Πχ Διαγραφή του 50 με successor (60)

Πρώτα ανεβάζουμε το 70 στη θέση του 60, γιατί το 60 είναι πατέρας με ένα παιδί

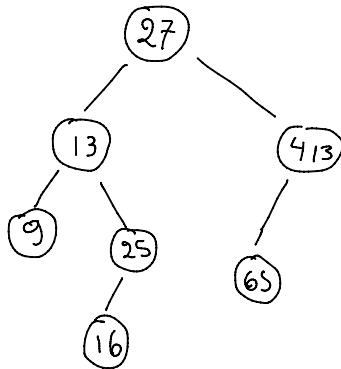


Αλλάζουμε το 50 με 60

①

65, 9, 13, 16, 25, 27, 413

9, 13, 16, 25, 27, 65, 413
✓ ✓ ✓ ✓ ✓ ✓



Function3(root, 25) == 25

Function3(root, 67) == 65

```
Function3(BinTreeNode *Root, ---2---)
{

}
```