



Δομές Δεδομένων

Ενότητα 5: Δυαδικά Δένδρα

Απόστολος Παπαδόπουλος
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





Δυαδικά Δένδρα



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα ενότητας

1. Δυαδικά Δένδρα (binary trees).
2. Δυαδικά Δένδρα Αναζήτησης (binary search trees).
3. Ισοζυγισμένα Δυαδικά Δένδρα (balanced binary trees).
4. Δένδρα Αναζήτησης m -δρόμων.
5. Πολυδιάστατα Δένδρα (multidimensional trees).



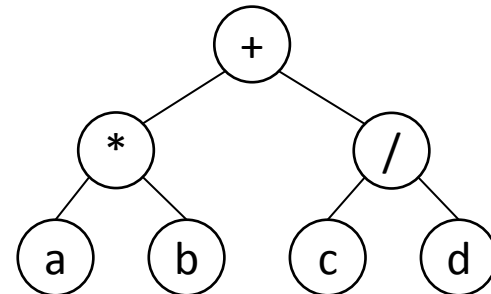
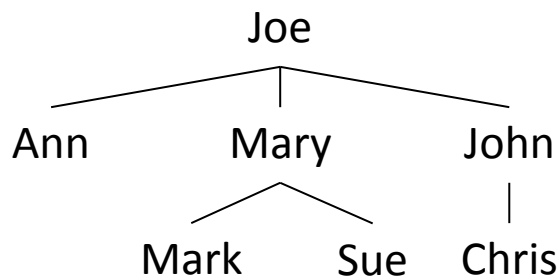


ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Δυαδικά Δένδρα

Ορισμοί (1/3)

- (Αναδρομικός ορισμός) - Ένα δένδρο t είναι ένα πεπερασμένο μη κενό σύνολο στοιχείων. Ένα από τα στοιχεία αυτά ονομάζεται ρίζα, ενώ τα υπόλοιπα στοιχεία (αν υπάρχουν) επιμερίζονται σε δένδρα που ονομάζονται υποδένδρα του t .
 - Βαθμός ενός στοιχείου είναι ο αριθμός των παιδιών που έχει.
 - Βαθμός του δένδρου είναι ο μέγιστος βαθμός των στοιχείων του.



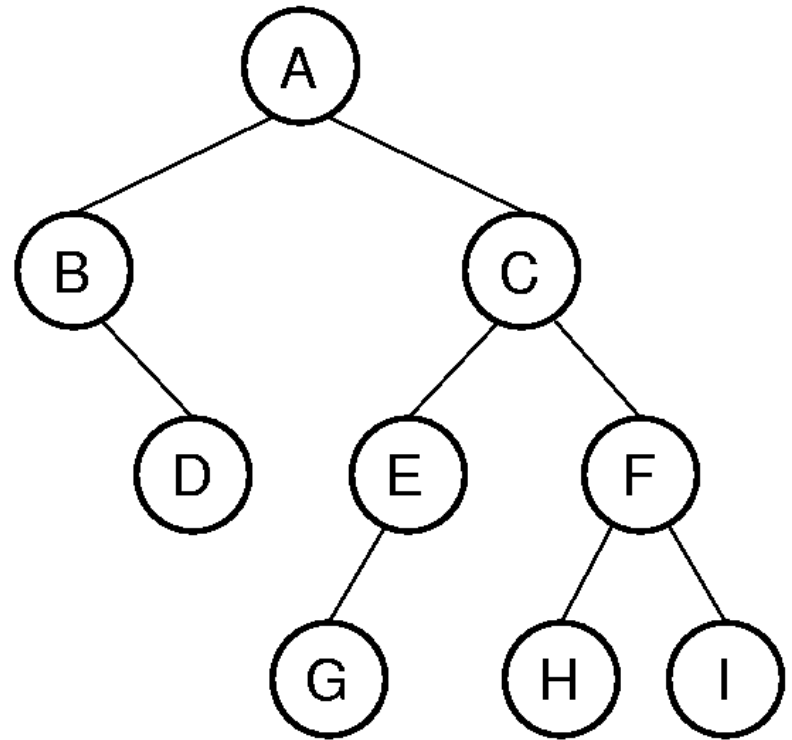
Ορισμοί (2/3)

- (Αναδρομικός ορισμός) - Ένα δυαδικό δένδρο t είναι μία πεπερασμένη (πιθανώς κενή) συλλογή στοιχείων. Όταν το δυαδικό δένδρο δεν είναι κενό, τότε έχει μία ρίζα και τα υπόλοιπα στοιχεία (αν υπάρχουν) επιμερίζονται σε δύο δυαδικά δένδρα που ονομάζονται το αριστερό και το δεξιό υποδένδρο του t .
- Ιδιότητες:
 - Το σχέδιο ενός δυαδικού δένδρου με n στοιχεία ($n > 0$) έχει ακριβώς $n-1$ ακμές.
 - Ένα δυαδικό δένδρο ύψους h ($h \geq 0$) έχει τουλάχιστον h και το πολύ 2^h-1 στοιχεία.



Ορισμοί (3/3)

- στοιχείο (κόμβος)
- ακμή
- υποδένδρο
- παιδιά, γονιός, πρόγονος, απόγονος
- μονοπάτι
- ύψος (βάθος)
- επίπεδο
- φύλλα, εσωτερικοί κόμβοι



Πλήρη και Συμπληρωμένα Δυαδικά Δένδρα

- Πλήρες (full) δυαδικό δένδρο ύψους h είναι εκείνο το δυαδικό δένδρο, το οποίο περιέχει ακριβώς $2^h - 1$ στοιχεία.
 - Μπορούμε να αριθμήσουμε από 1 έως $2^h - 1$ τα στοιχεία ενός πλήρους δυαδικού δένδρου ύψους h , ξεκινώντας από το επίπεδο 1 προς το επίπεδο h και, μέσα σε κάθε επίπεδο, από αριστερά προς τα δεξιά.
- Συμπληρωμένο (complete) δυαδικό δένδρο ύψους h είναι εκείνο το δυαδικό δένδρο, το οποίο προκύπτει από ένα πλήρες δυαδικό δένδρο ύψους h εάν διαγράψουμε k στοιχεία με αρίθμηση $2^h - i$, για $1 \leq i \leq k$ ($k \geq 0$).



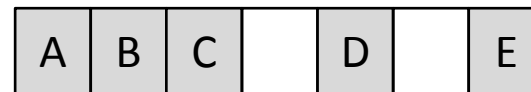
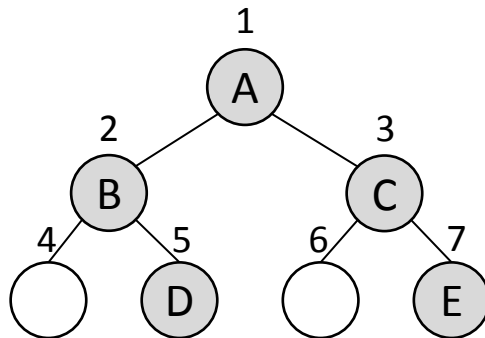
Ιδιότητες πλήρων / συμπληρωμένων δυναδικών δένδρων

- Το πλήθος των φύλλων σε ένα μη άδειο πλήρες δυναδικό δένδρο είναι κατά 1 μεγαλύτερο από το πλήθος των εσωτερικών κόμβων ($2^h - 1$ και $2^{h-1} - 1$, αντίστοιχα).
- Έστω i ($1 \leq i \leq n$) ο αριθμός ενός στοιχείου ενός συμπληρωμένου δυναδικού δένδρου:
 - Αν $i = 1$, το στοιχείο είναι η ρίζα, αλλιώς είναι παιδί του κόμβου με αριθμό $\lfloor i/2 \rfloor$.
 - Το αριστερό του παιδί έχει αριθμό $2i$ (αν $2i \leq n$, αλλιώς δεν έχει αριστερό παιδί).
 - Το δεξιό του παιδί έχει αριθμό $2i+1$ (αν $2i+1 \leq n$, αλλιώς δεν έχει δεξιό παιδί).
- Αποδείξεις με μαθηματική επαγωγή.

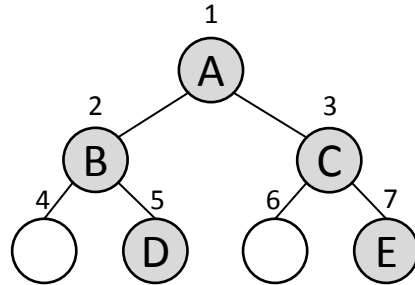


Στατική αναπαράσταση δυαδικού δένδρου (με πίνακα)

- Βασίζεται στην προηγούμενη ιδιότητα (των συμπληρωμένων δυαδικών δένδρων).
- Πρόβλημα: σπατάλη χώρου όταν λείπουν πολλά στοιχεία (για να γίνει το δένδρο πλήρες).
 - Ένα δένδρο με n στοιχεία θα μπορούσε να απαιτήσει πίνακα μεγέθους μέχρι και $2^n - 1$ (η περίπτωση των δεξιών λοξών δυαδικών δένδρων).



Υλοποίηση δυαδικού δένδρου με πίνακα



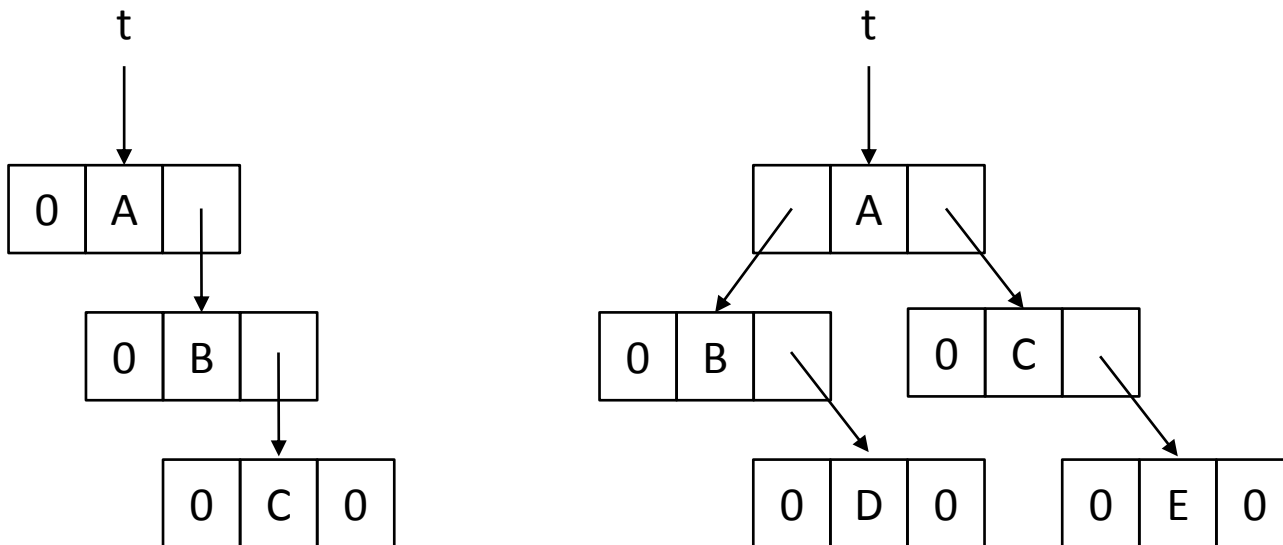
Θέση	1	2	3	4	5	6	7
Γονιός	--	1	1	2	2	3	3
Αριστερό παιδί	2	4	6	--	--	--	--
Δεξί παιδί	3	5	7	--	--	--	--
Αριστερός αδελφός	--	--	2	--	4	--	6
Δεξιός αδελφός	--	3	--	5	--	7	--

(οι μαθηματικοί τύποι προκύπτουν από την ιδιότητα των συμπληρωμένων δυαδικών δένδρων).



Συνδεδεμένη αναπαράσταση δυαδικού δένδρου (με δείκτες)

- Η πιο δημοφιλής υλοποίηση.
- Κάθε στοιχείο αντιστοιχεί σε ένα κόμβο με ένα πεδίο δεδομένων (data) και δύο πεδία συνδέσμων (LeftChild, RightChild).



Κλάση κόμβου για συνδεδεμένα δυναμικά δένδρα

```
class BinaryTreeNode {  
    public:  
        BinaryTreeNode() {LeftChild = RightChild = 0;}  
        BinaryTreeNode(const T& e)  
            {data = e; LeftChild = RightChild = 0;}  
        BinaryTreeNode(const T& e, BinaryTreeNode *l,  
            BinaryTreeNode *r)  
            {data = e; LeftChild = l; RightChild = r;}  
private:  
    T data;  
    BinaryTreeNode<T> *LeftChild, // left subtree  
        *RightChild; // right subtree  
}
```



Πράξεις πάνω σε δυαδικά δένδρα

- Προσδιορισμός ύψους, πλήθους στοιχείων.
- Δημιουργία αντιγράφου.
- Παρουσίαση δένδρου σε μονάδα εξόδου.
- Διαγραφή δένδρου.
- Υπολογισμός έκφρασης (αν είναι δένδρο έκφρασης).

Όλα τα παραπάνω εκτελούνται με συστηματικό τρόπο με τη λειτουργία διάσχισης του δένδρου.



Διάσχιση δένδρου (1/3)

- Κάθε διαδικασία επίσκεψης όλων των κόμβων ενός δένδρου, ακριβώς μια φορά τον καθένα, ονομάζεται διάσχιση (traversal).
 - Προδιατεταγμένη διάσχιση (preorder): 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 διάσχιση (postorder): 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.



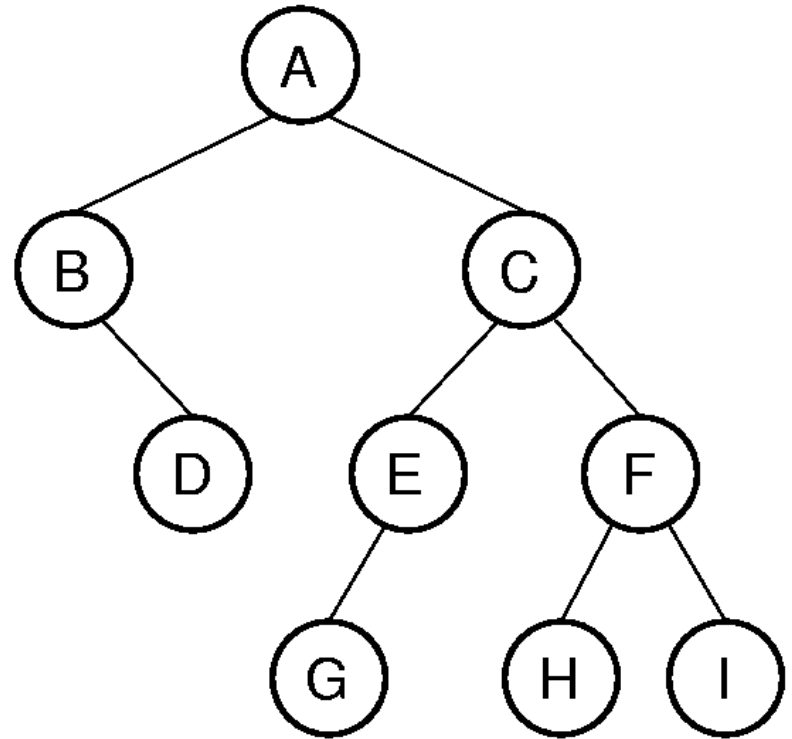
Διάσχιση δένδρου (2/3)

- **Ενδοδιατεταγμένη** διάσχιση (inorder): 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** (level order): Επισκεπτόμαστε τους κόμβους κατά επίπεδα, από πάνω (τη ρίζα) προς τα κάτω, και μέσα σε ένα επίπεδο από αριστερά προς τα δεξιά.
- Οι 3 πρώτες μέθοδοι είναι αναδρομικές ενώ η 4η είναι επαναληπτική.



Διάσχιση δένδρου (3/3)

- Προδιατεταγμένη:
A, B, D, C, E, G, F, H, I
- Μεταδιατεταγμένη:
D, B, G, E, H, I, F, C, A
- Ενδοδιατεταγμένη:
B, D, A, G, E, C, H, F, I
- Κατά σειρά επιπέδων:
A, B, C, D, E, F, G, H, I



Αναδρομικές μέθοδοι διάσχισης

```
void PreOrder(BinaryTreeNode<T> *t)
{
  // Preorder traversal of *t.
  if (t) {Visit(t); PreOrder(t->LeftChild);
    PreOrder(t->RightChild); }
}

void InOrder(BinaryTreeNode<T> *t)
{
  // Inorder traversal of *t.
  if (t) {InOrder(t->LeftChild);
    Visit(t); InOrder(t->RightChild); }
}

void PostOrder(BinaryTreeNode<T> *t)
{
  // Postorder traversal of *t.
  if (t) {PostOrder(t->LeftChild);
    PostOrder(t->RightChild); Visit(t); }
}
```



Διάσχιση κατά σειρά επιπέδων

```
void LevelOrder(BinaryTreeNode<T> *t)
{
    // Level-order traversal of *t.
    LinkedQueue<BinaryTreeNode<T>*> Q;
    while (t) {
        Visit(t); // visit t
        // put t's children on queue
        if (t->LeftChild) Q.Add(t->LeftChild);
        if (t->RightChild) Q.Add(t->RightChild);
        // get next node to visit
        try {Q.Delete(t);}
        catch (OutOfBounds) {return;}
    }
}
```



ΑΤΔ BinaryTree

AbstractDataType *BinaryTree* {

instances: collection of elements; if not empty, the collection is partitioned into a root, left subtree, and right subtree; each subtree is also a binary tree;

operations

Create (): create an empty binary tree

IsEmpty (): return true if the tree is empty, return false otherwise

Root (x): x is set to root element; return false if the operation fails,
return true otherwise

MakeTree (root, left, right): create a binary tree with root as the root
element, left (right) as the left (right) subtree.

BreakTree (root, left, right): inverse of create

PreOrder: preorder traversal of binary tree

InOrder: inorder traversal of binary tree

PostOrder: postorder traversal of binary tree

LevelOrder: level-order traversal of binary tree

}



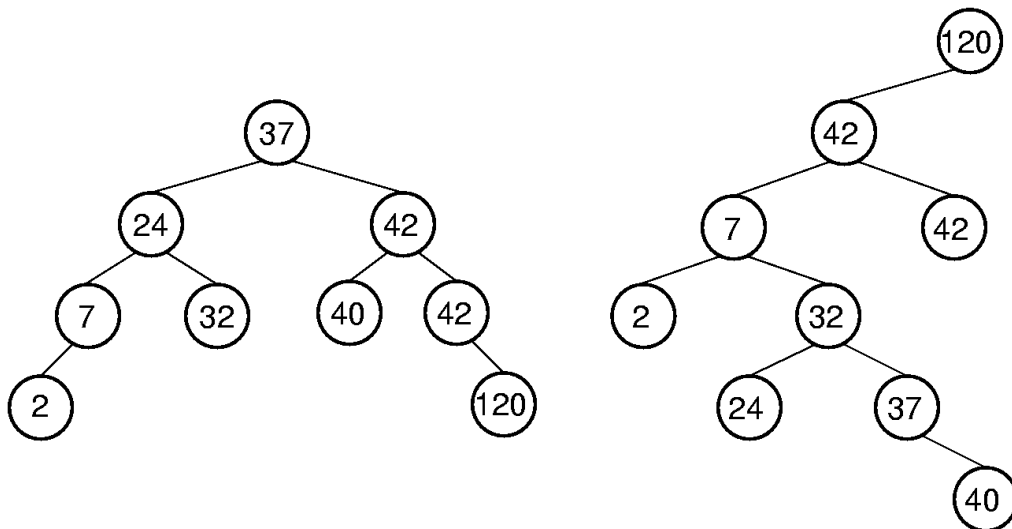


ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Δυαδικά Δένδρα Αναζήτησης

Δυαδικό Δένδρο Αναζήτησης (ΔΔΑ)

- Ορισμός: Ένα ΔΔΑ είναι δυαδικό δένδρο με διακριτά κλειδιά (τιμές) και τις εξής ιδιότητες:
 - Τα κλειδιά (αν υπάρχουν) στο αριστερό υποδένδρο της ρίζας είναι μικρότερα από το κλειδί της ρίζας.
 - Τα κλειδιά (αν υπάρχουν) στο δεξιό υποδένδρο της ρίζας είναι μεγαλύτερα από το κλειδί της ρίζας.
 - Το αριστερό και το δεξιό υποδένδρο είναι επίσης ΔΔΑ.



Κίνητρο:

**να μειώσουμε τους
χρόνους
ενημέρωσης και
αναζήτησης σε
λιγότερο από $\Theta(n)$**



ΑΤΔ BinarySearchTree

AbstractDataType *BSTree* {

instances: binary trees, each node has an element with a key field;
all keys are distinct; keys in the left subtree of any node smaller
than the key in the node; those in the right subtree are larger;

operations

Create (): create an empty binary search tree

Search (*k*, *e*): return in *e* the element with key *k*; return false if
the operation fails, return true if it succeeds

Insert (*e*): insert element *e* into the search tree

Delete (*k*, *e*): delete the element with key *k* and return it in *e*

Ascend (): Output all elements in ascending order of key

}



Αναζήτηση στοιχείου μέσα σε ΔΔΑ

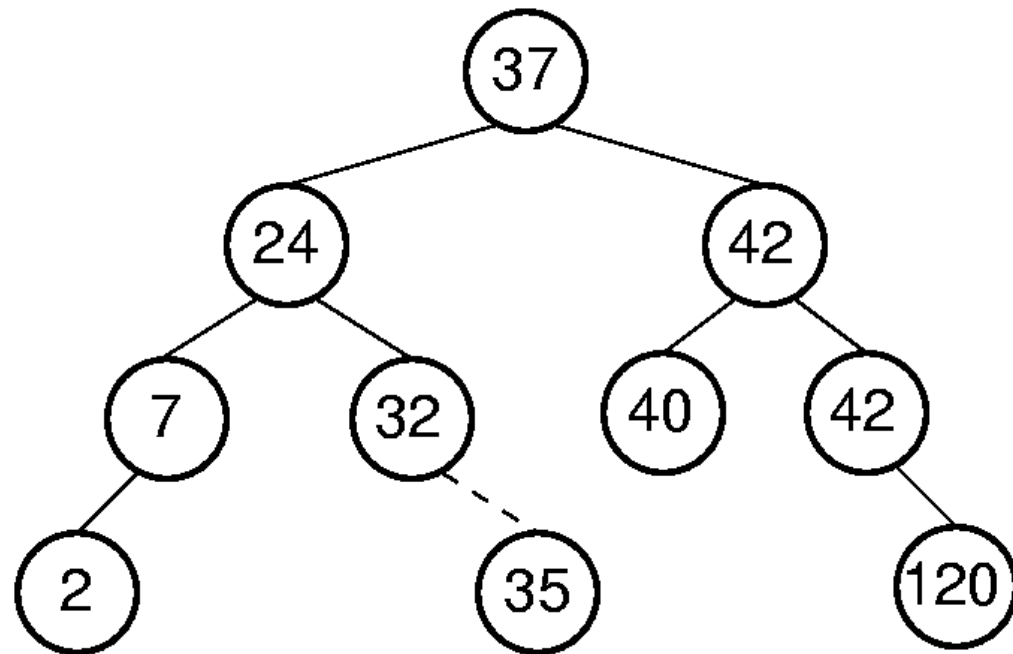
```
bool BSTree<E,K>::Search(const K& k, E &e) const
{// Search for element that matches k.
 // pointer p starts at the root and moves through
 // the tree looking for an element with key k
 BinaryTreeNode<E> *p = root;
 while (p) // examine p->data
   if (k < p->data) p = p->LeftChild;
   else if (k > p->data) p = p->RightChild;
   else { // found element
     e = p->data;
     return true;}
 return false;
}
```

Κόστος αναζήτησης = κόστος κατάβασης = $O(h)$



Εισαγωγή στοιχείου σε ΔΔΑ (1/4)

- Βασική ιδιότητα ΔΔΑ:
 - Η εισαγωγή γίνεται πάντα σε κάποιο (νέο) φύλλο.
- Διαδικασία:
 - Αναζήτηση του στοιχείου (οπότε καταλήγουμε σε κόμβο-φύλλο).
 - Εισαγωγή του ως παιδί εκείνου του κόμβου.



Εισαγωγή στοιχείου σε ΔΔΑ (2/4)

```
BSTree<E,K>& BSTree<E,K>::Insert(const E& e)
{
    // Insert e if not duplicate.
    BinaryTreeNode<E> *p = root, // search pointer
                    *pp = 0; // parent of p
    // find place to insert
    while (p) { // examine p->data
        pp = p;
        // move p to a child
        if (e < p->data) p = p->LeftChild;
        else if (e > p->data) p = p->RightChild;
        else throw BadInput(); // duplicate
    }

    // get a node for e and attach to pp
    ...
}
```



Εισαγωγή στοιχείου σε ΔΔΑ (3/4)

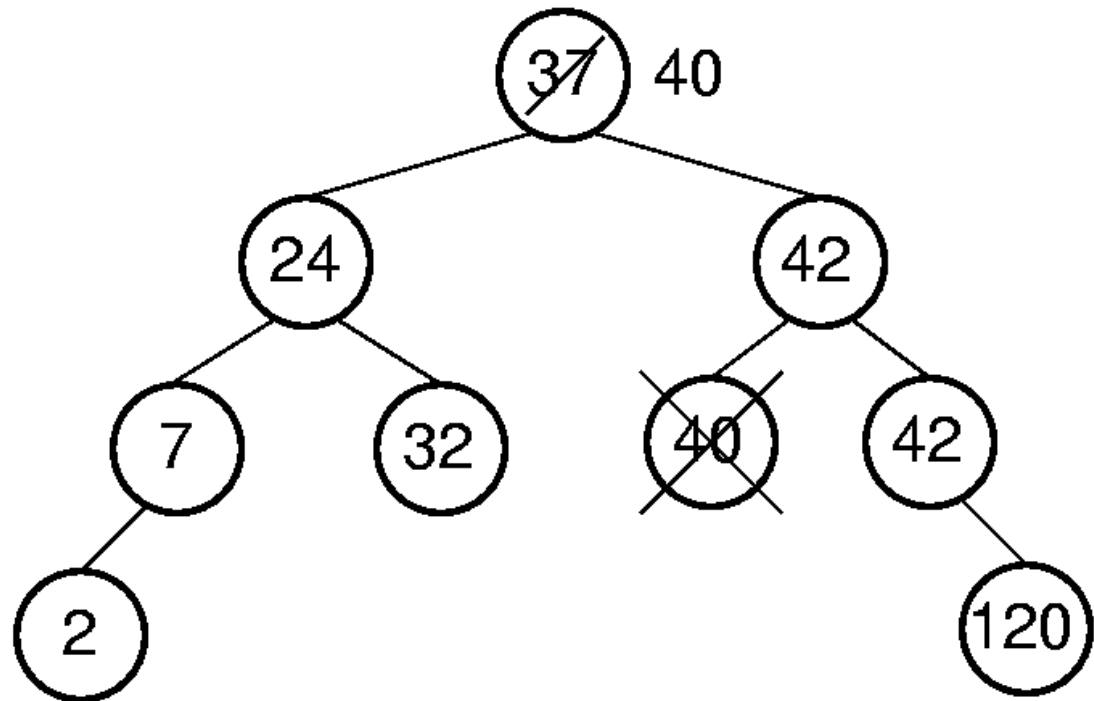
```
...  
// get a node for e and attach to pp  
BinaryTreeNode<E> *r = new BinaryTreeNode<E> (e);  
if (root) { // tree not empty  
    if (e < pp->data) pp->LeftChild = r;  
    else pp->RightChild = r;}  
else // insertion into empty tree  
    root = r;  
  
return *this;
```

} **Κόστος = Κόστος αναζήτησης + κόστος 'συγκόλλησης'
νέου κόμβου στον πατέρα-κόμβο = $O(h) + O(1) = O(h)$**



Διαγραφή στοιχείου από ΔΔΑ (4/4)

- 3 περιπτώσεις:
 - Ο κόμβος p (που περιέχει το στοιχείο) είναι φύλλο
 - Το p έχει μόνο ένα μη κενό υποδένδρο
 - Το p έχει ακριβώς δύο μη κενά υποδένδρα



Διαγραφή στοιχείου από ΔΔΑ (1/3)

```
BSTree<E,K>& BSTree<E,K>::Delete(const K& k, E& e)
{
  // Delete element with key k and put it in e.
  // set p to point to node with key k
  BinaryTreeNode<E> *p = root, // search pointer
    *pp = 0; // parent of p
  while (p && p->data != k){ // move to a child of p
    pp = p;
    if (k < p->data) p = p->LeftChild;
    else p = p->RightChild;
  }
  if (!p) throw BadInput(); // no element with key k
  e = p->data; // save element to delete

```

...



Διαγραφή στοιχείου από ΔΔΑ (2/3)

...

```
// restructure tree
// handle case when p has two children
if (p->LeftChild && p->RightChild) { // two children
    // convert to zero or one child case
    // find largest element in left subtree of p
    BinaryTreeNode<E> *s = p->LeftChild,
        *ps = p; // parent of s
    while (s->RightChild) { // move to larger element
        ps = s;
        s = s->RightChild;}
    // move largest from s to p
    p->data = s->data;
    p = s;
    pp = ps;}
}
```

...



Διαγραφή στοιχείου από ΔΔΑ (3/3)

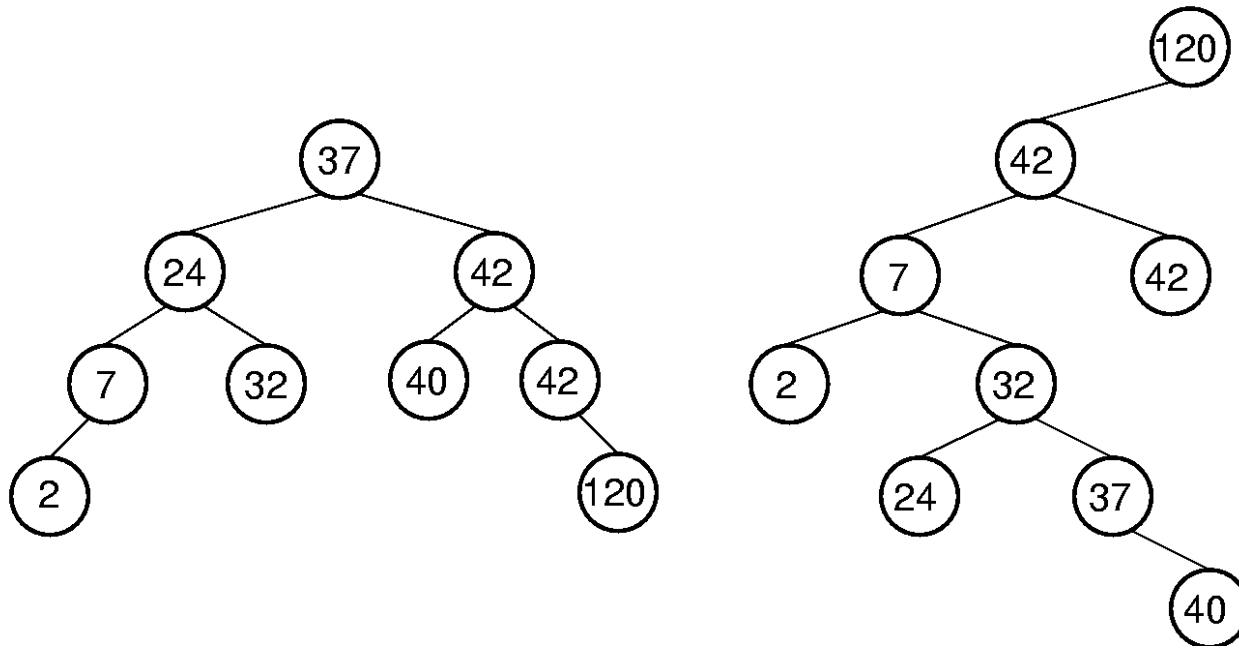
...

```
// p has at most one child
// save child pointer in c
BinaryTreeNode<E> *c;
if (p->LeftChild) c = p->LeftChild;
else c = p->RightChild;
// delete p
if (p == root) root = c;
else { // is p left or right child of pp?
    if (p == pp->LeftChild)
        pp->LeftChild = c;
    else pp->RightChild = c;}
delete p;
return *this;
}
```



Ύψος ΔΔΑ

- Το ύψος ενός ΔΔΑ με n στοιχεία μπορεί να φτάσει μέχρι και n .
- Στη γενική περίπτωση όμως (όταν οι εισαγωγές και οι διαγραφές γίνονται τυχαία), το ύψος είναι $O(\log n)$ κατά μ.ό.

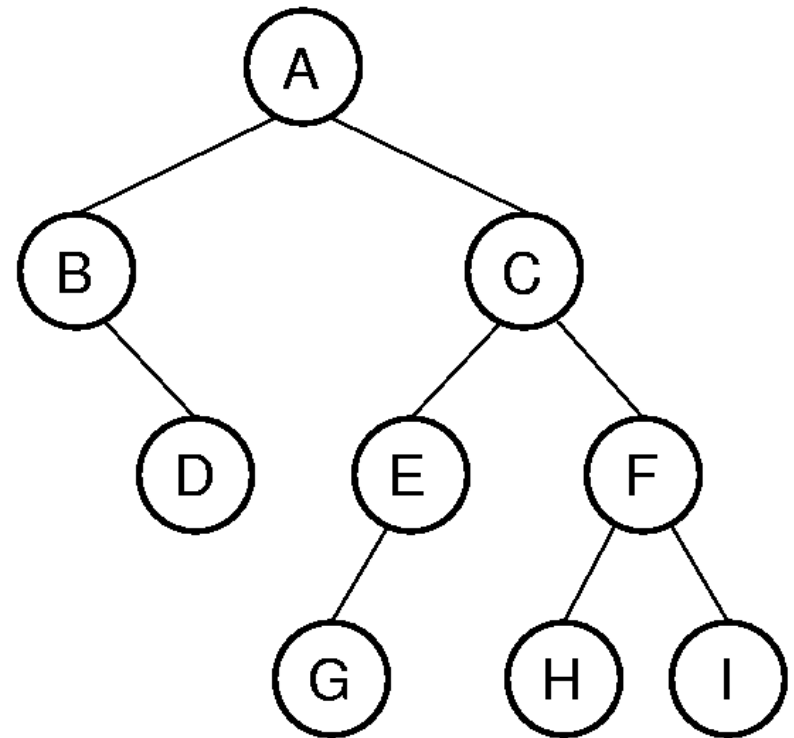




Ισοζυγισμένα Δυαδικά Δένδρα, Δένδρα Αναζήτησης m -δρόμων, Πολυδιάστατα Δένδρα

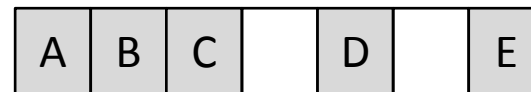
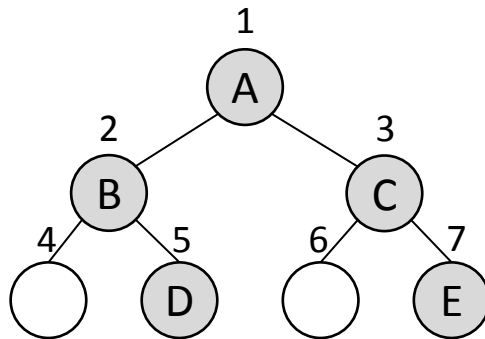
Ορισμοί

- στοιχείο (κόμβος)
- ακμή
- υποδένδρο
- παιδιά, γονιός, πρόγονος, απόγονος
- μονοπάτι
- ύψος (βάθος)
- επίπεδο
- φύλλα, εσωτερικοί κόμβοι



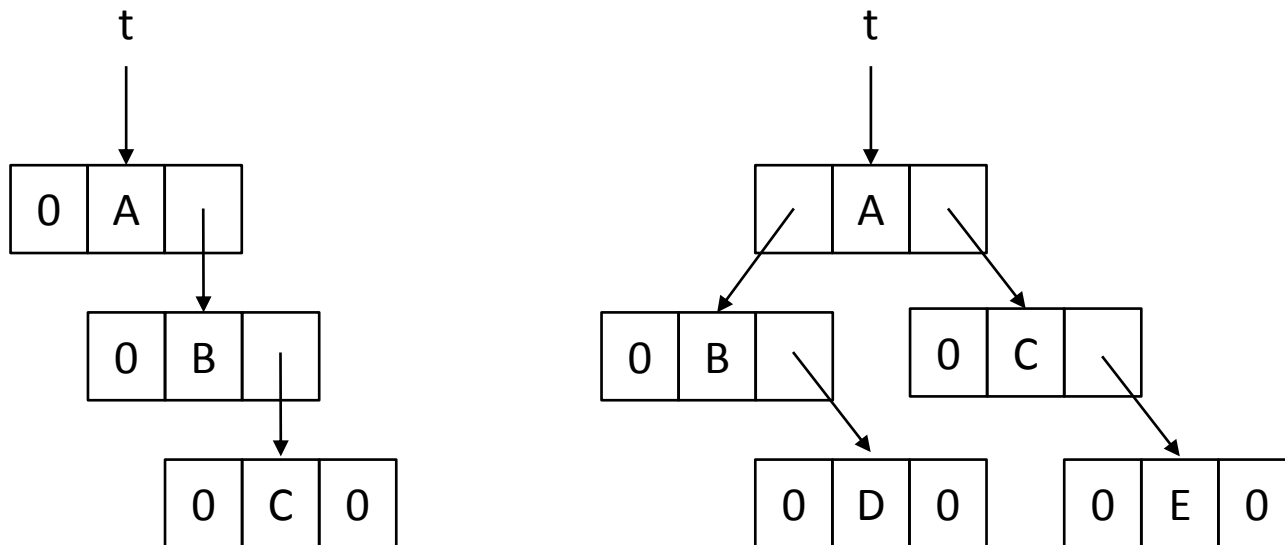
Στατική αναπαράσταση δυαδικού δένδρου (με πίνακα)

- Βασίζεται στην προηγούμενη ιδιότητα (των συμπληρωμένων δυαδικών δένδρων)
- Πρόβλημα: σπατάλη χώρου όταν λείπουν πολλά στοιχεία (για να γίνει το δένδρο πλήρες)
 - Ένα δένδρο με n στοιχεία θα μπορούσε να απαιτήσει πίνακα μεγέθους μέχρι και $2^n - 1$ (η περίπτωση των δεξιών λοξών δυαδικών δένδρων)



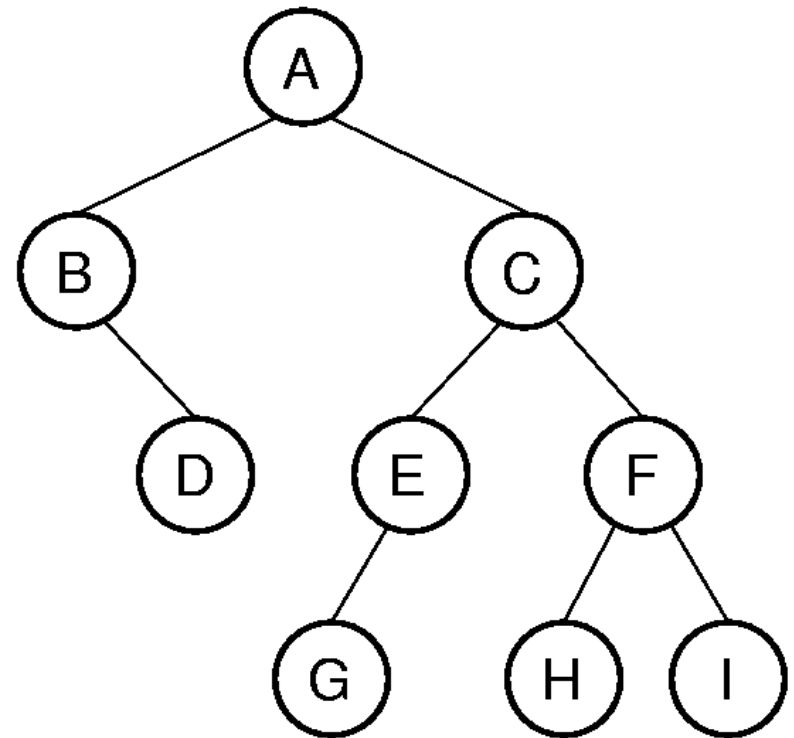
Συνδεδεμένη αναπαράσταση δυαδικού δένδρου (με δείκτες)

- Η πιο δημοφιλής υλοποίηση
- Κάθε στοιχείο αντιστοιχεί σε ένα κόμβο με ένα πεδίο δεδομένων (data) και δύο πεδία συνδέσμων (LeftChild, RightChild)



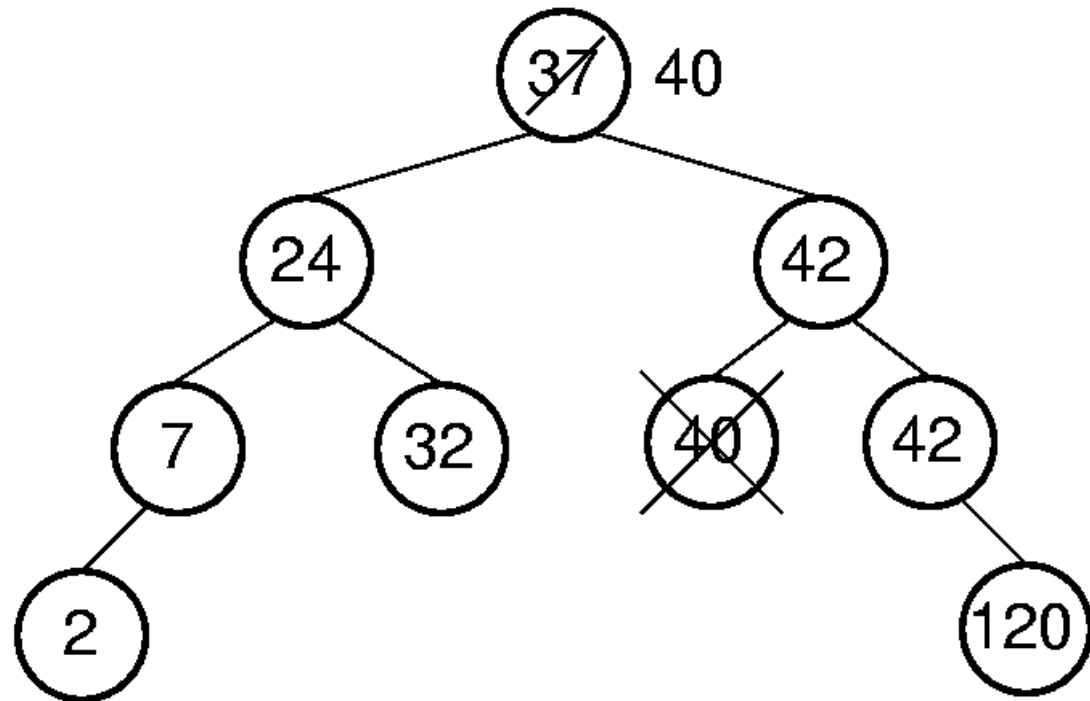
Διάσχιση δένδρου

- Προδιατεταγμένη:
A, B, D, C, E, G, F, H, I
- Μεταδιατεταγμένη:
D, B, G, E, H, I, F, C, A
- Ενδοδιατεταγμένη:
B, D, A, G, E, C, H, F, I
- Κατά σειρά
επιπέδων:
A, B, C, D, E, F, G, H, I

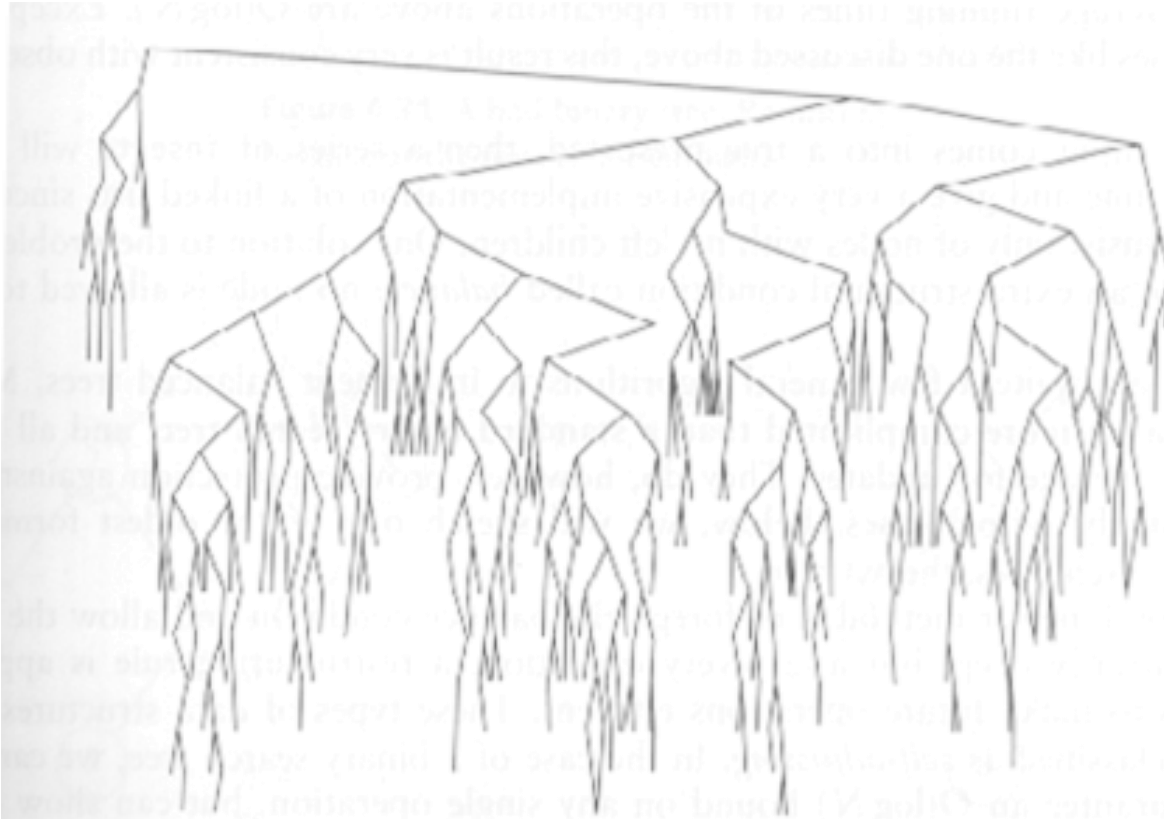


Διαγραφή στοιχείου από ΔΔΑ

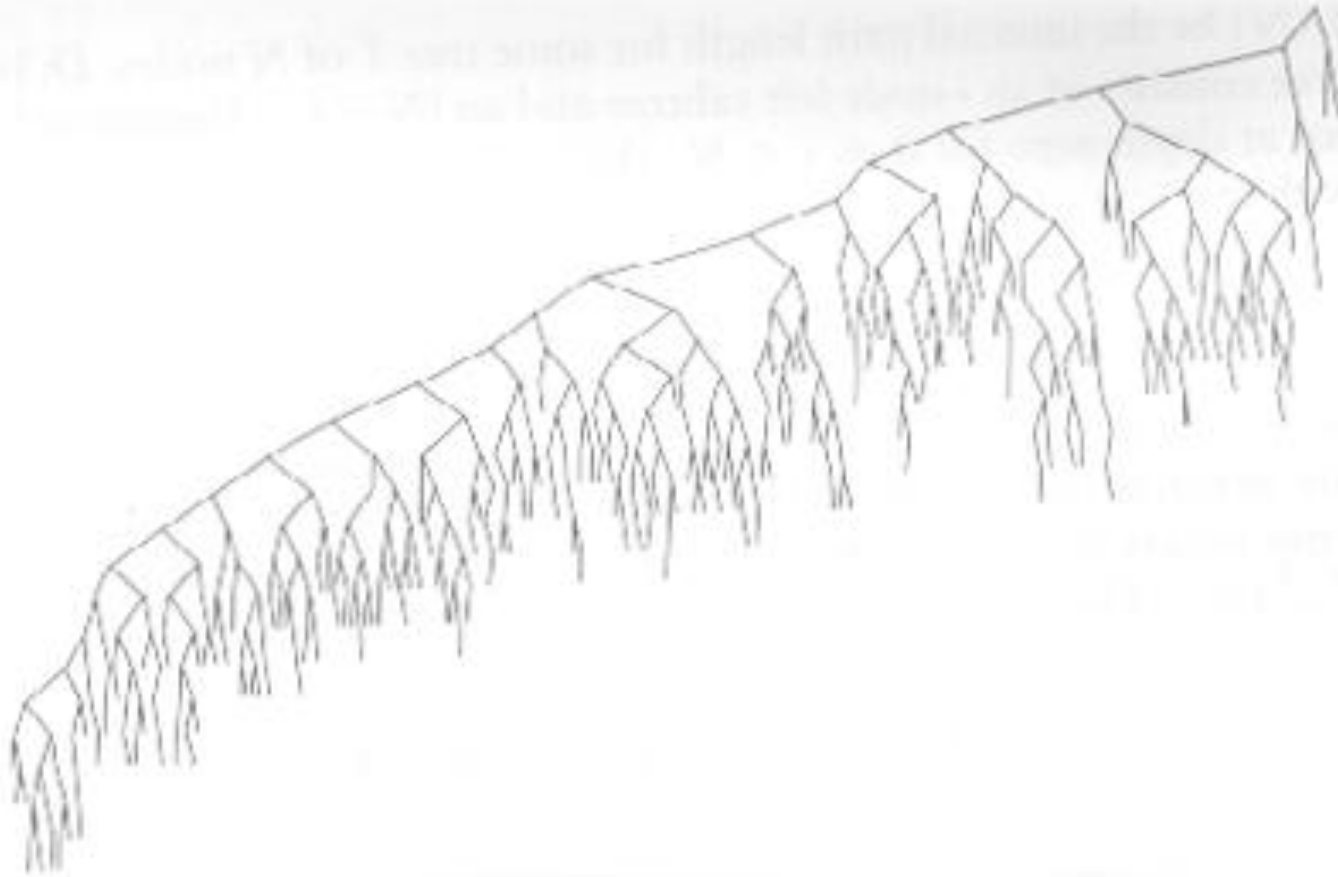
- 3 περιπτώσεις:
 - Ο κόμβος p (που περιέχει το στοιχείο) είναι φύλλο
 - Το p έχει μόνο ένα μη κενό υποδένδρο
 - Το p έχει ακριβώς δύο μη κενά υποδένδρα



Τυχαίο Δένδρο



Απαράδεκτο!



Ισοζυγισμένα Δένδρα

Στόχος:

- Να μην αφήσουμε το ένα κλαδί του δένδρου να γίνει κατά πολύ μεγαλύτερο του άλλου. Τα δύο υποδένδρα θέλουμε να έχουν περίπου το ίδιο ύψος.



Η Λύση

Δένδρα AVL

Adel'son

Vel'skii

Landis



Δένδρα AVL

- Η διαφορά των υψών του αριστερού και του δεξιού υποδένδρου δεν πρέπει να ξεπερνά το 1. Αυτό πρέπει να ισχύει αναδρομικά για όλους τους κόμβους του δένδρου.



Δένδρα AVL

- Τρόπος Λειτουργίας
- Η αναζήτηση είναι ίδια όπως και σε ένα απλό δυαδικό δένδρο αναζήτησης.
- Μετά από κάθε εισαγωγή και διαγραφή πρέπει να γίνει έλεγχος για το αν ικανοποιείται ο περιορισμός του AVL (διαφορά των υψών των υποδένδρων).



Δένδρα AVL

- Σε περίπτωση που δεν υπάρχει πρόβλημα στη δομή του δένδρου, τότε δεν απαιτείται καμία άλλη ενέργεια.
- Διαφορετικά θα πρέπει να γίνουν δομικές αλλαγές στο δένδρο ώστε να προκύψει πάλι ένα δένδρο AVL.

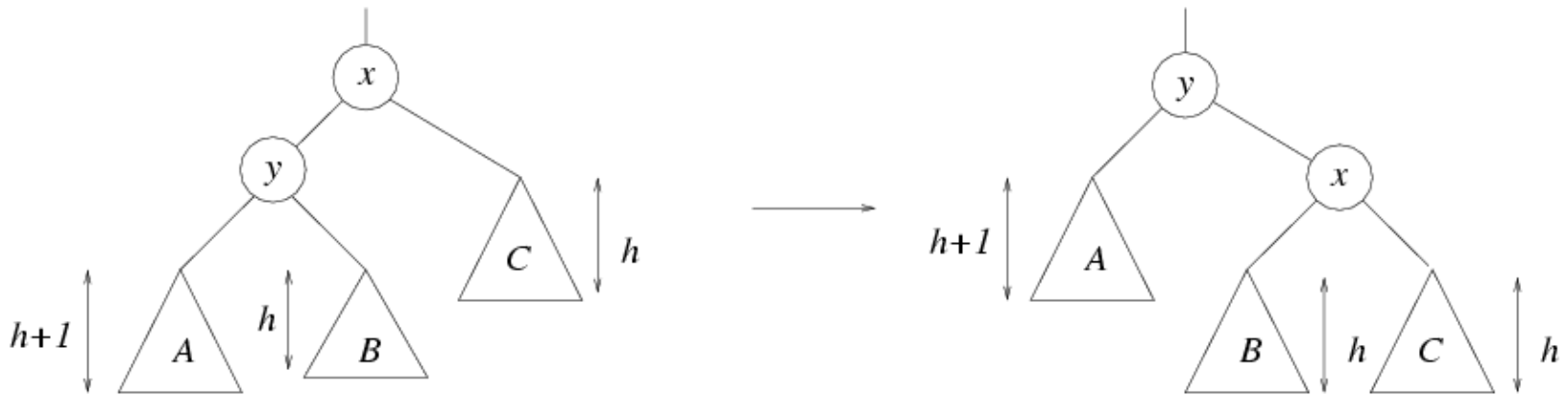


Περιστροφές

- Έχουμε δύο είδη περιστροφών
 - Απλή περιστροφή
 - Περιστρέφονται δύο κόμβοι
 - Διπλή περιστροφή
 - Περιστρέφονται τρεις κόμβοι
- Πρώτα θα δούμε τι είναι αυτές οι περιστροφές και στη συνέχεια θα εξετάσουμε τη χρήση τους

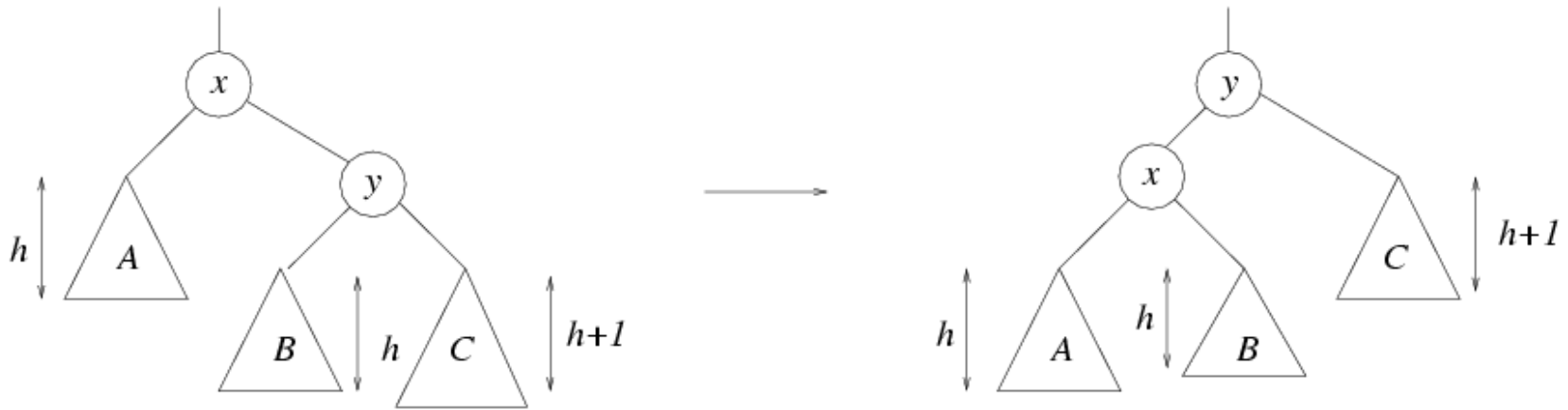


Απλή Περιστροφή



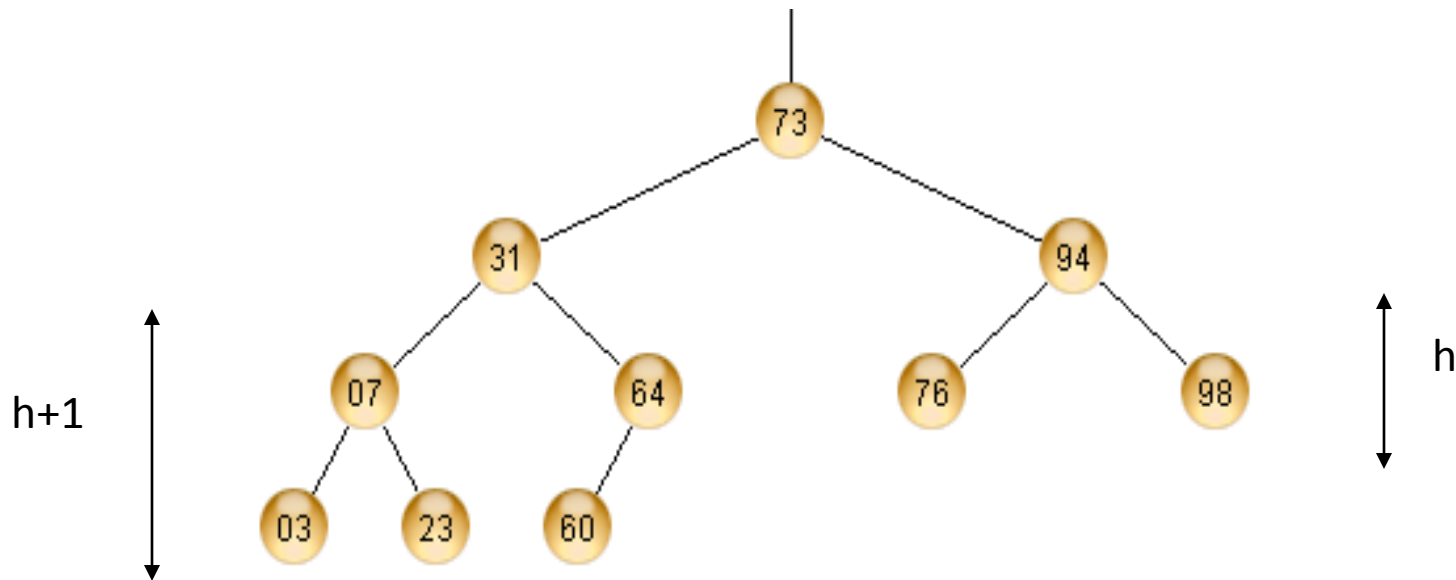
Περίπτωση 1

Απλή Περιστροφή



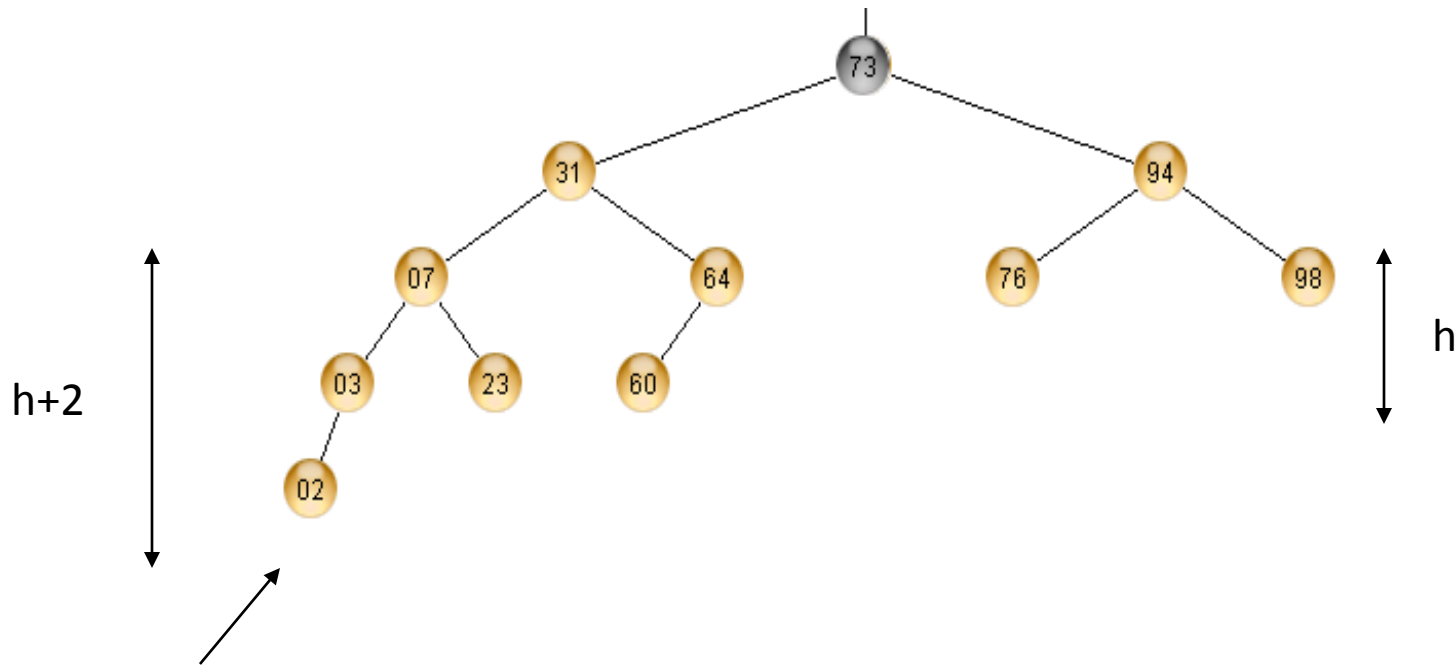
Περίπτωση 2

Απλή Περιστροφή - Παράδειγμα



Αυτό είναι ένα δένδρο AVL

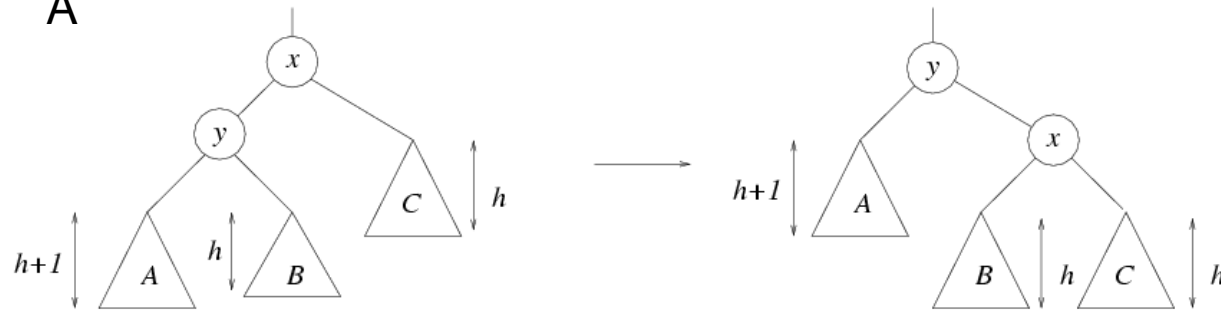
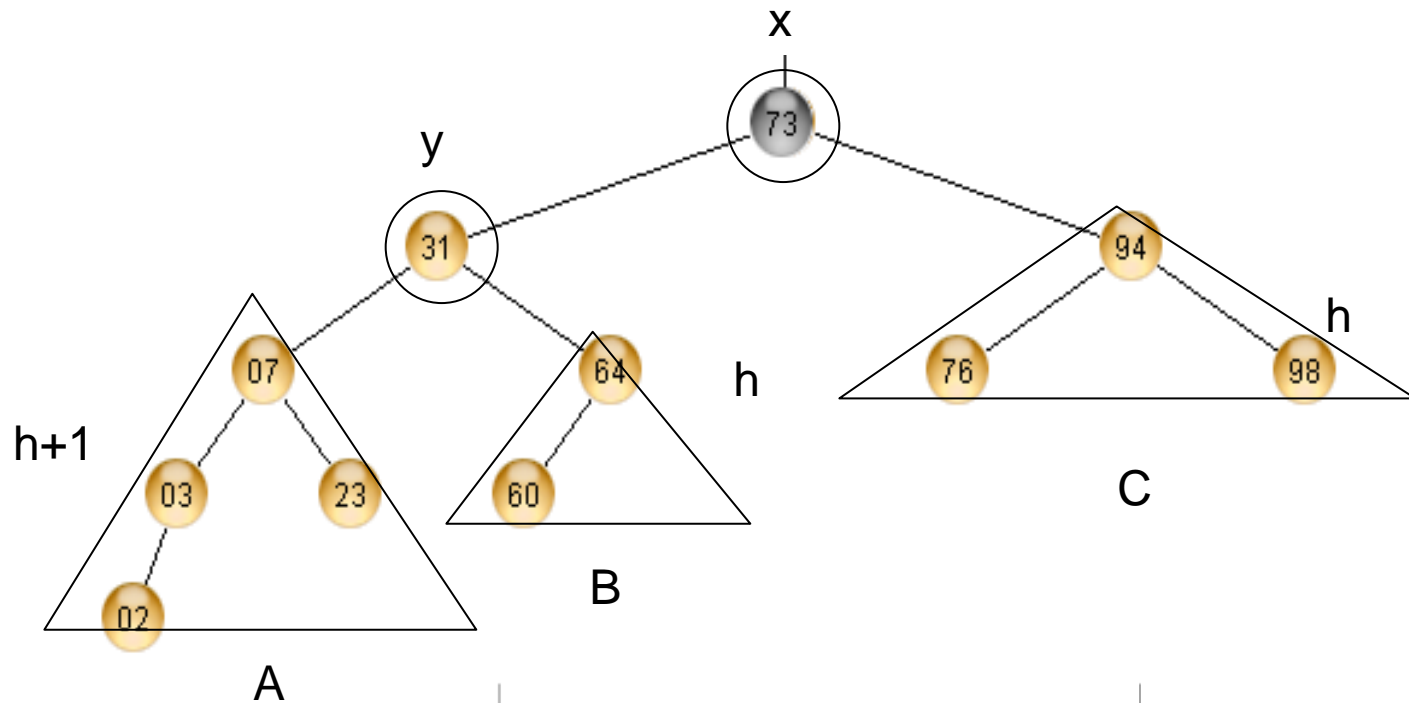
Απλή Περιστροφή - Παράδειγμα



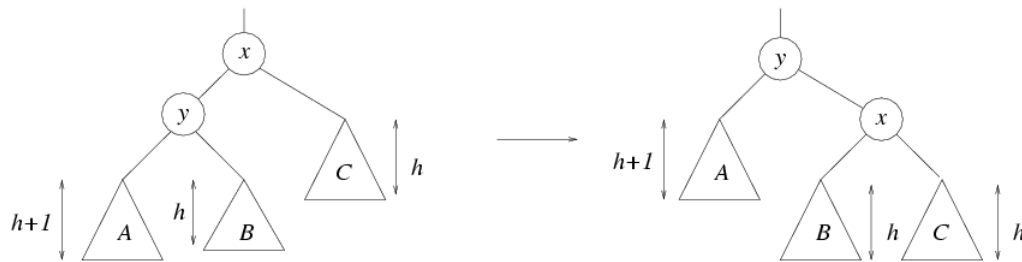
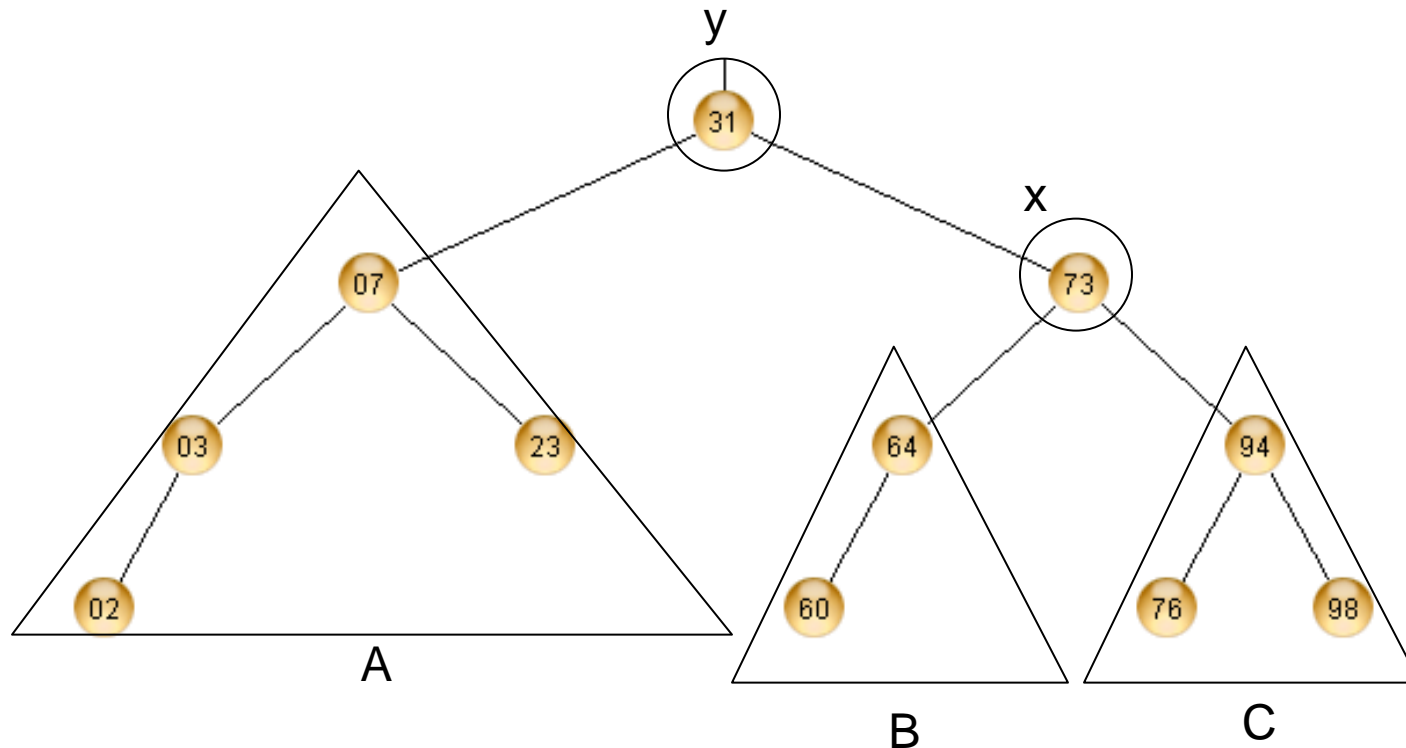
Εισαγωγή 02

Μετά την εισαγωγή του κόμβου 2 το δένδρο που προκύπτει ΔΕΝ είναι AVL

Απλή Περιστροφή - Παράδειγμα

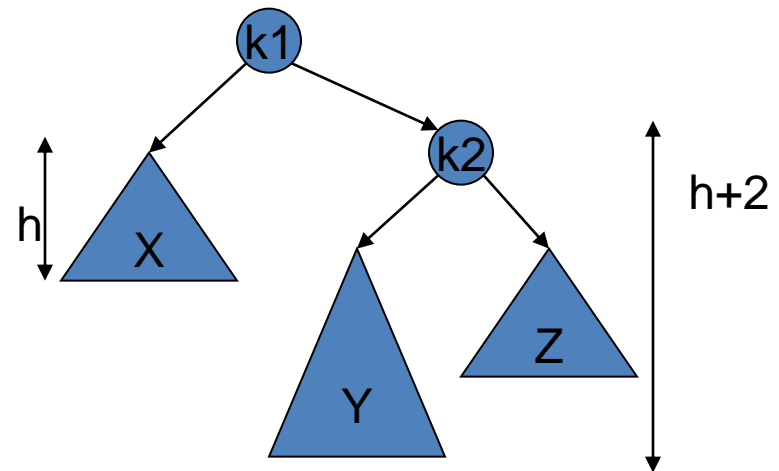
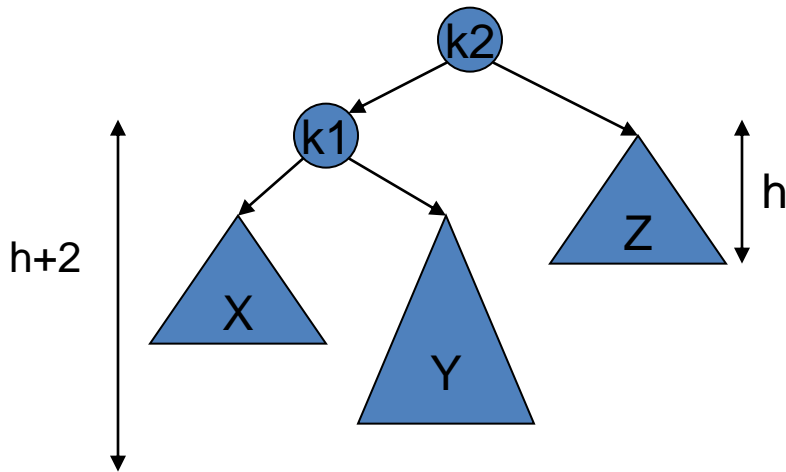


Μετά την Περιστροφή



Απλή Περιστροφή

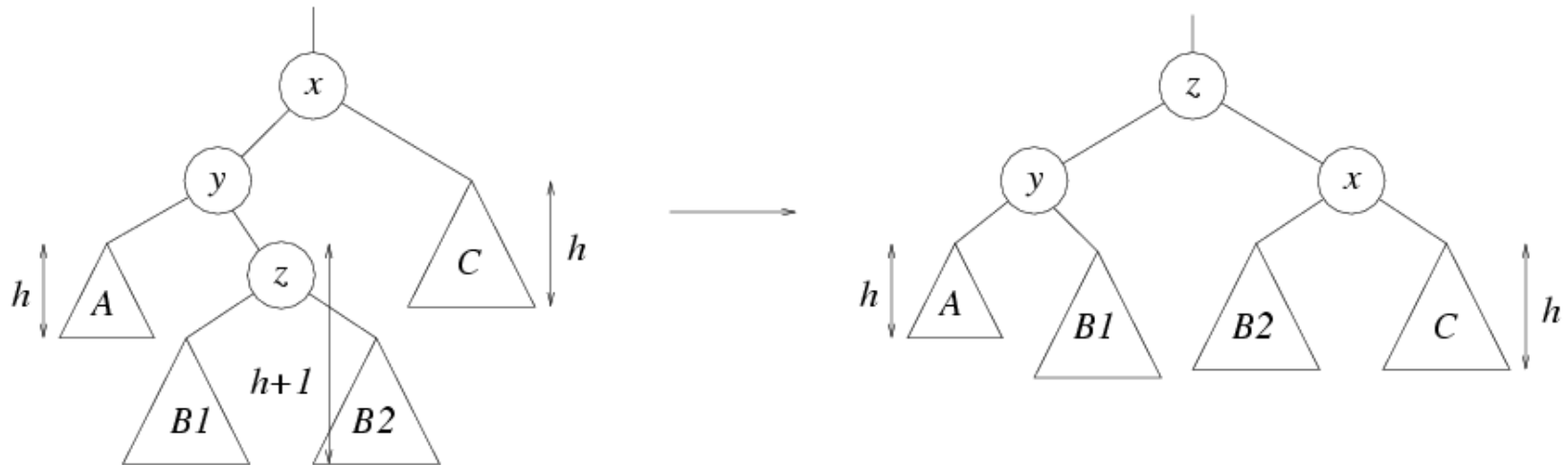
Μερικές φορές το πρόβλημα σε λύνεται με απλή περιστροφή



Τι μπορούμε να κάνουμε; Διπλή περιστροφή!

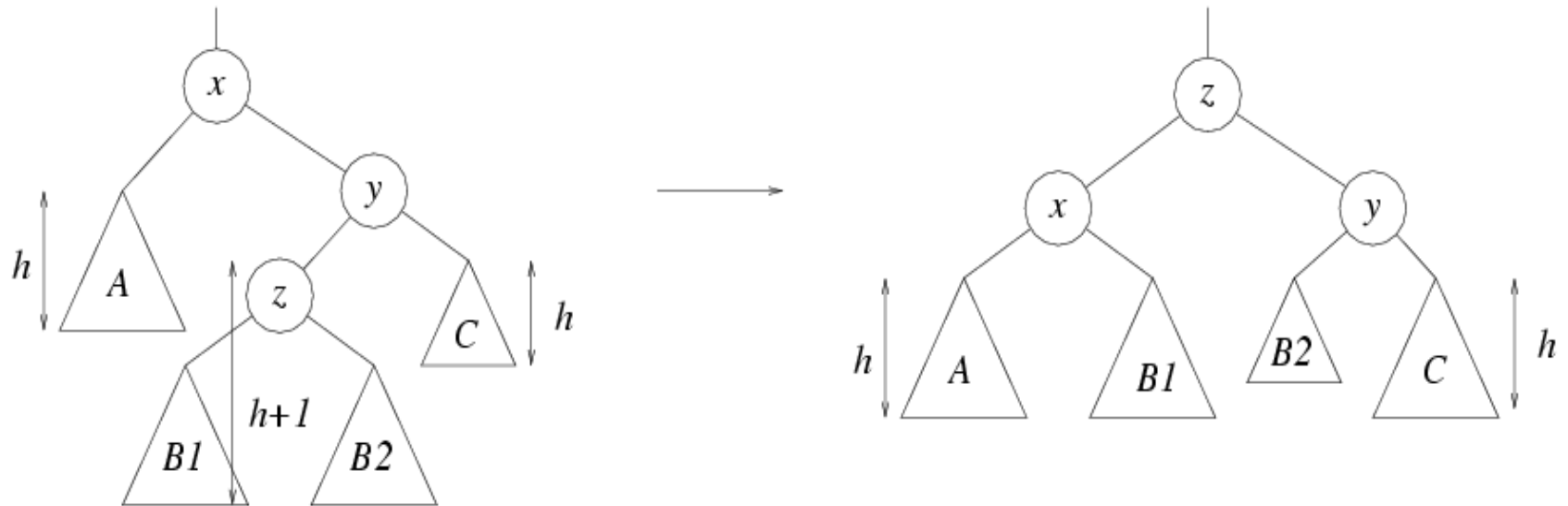


Διπλή Περιστροφή



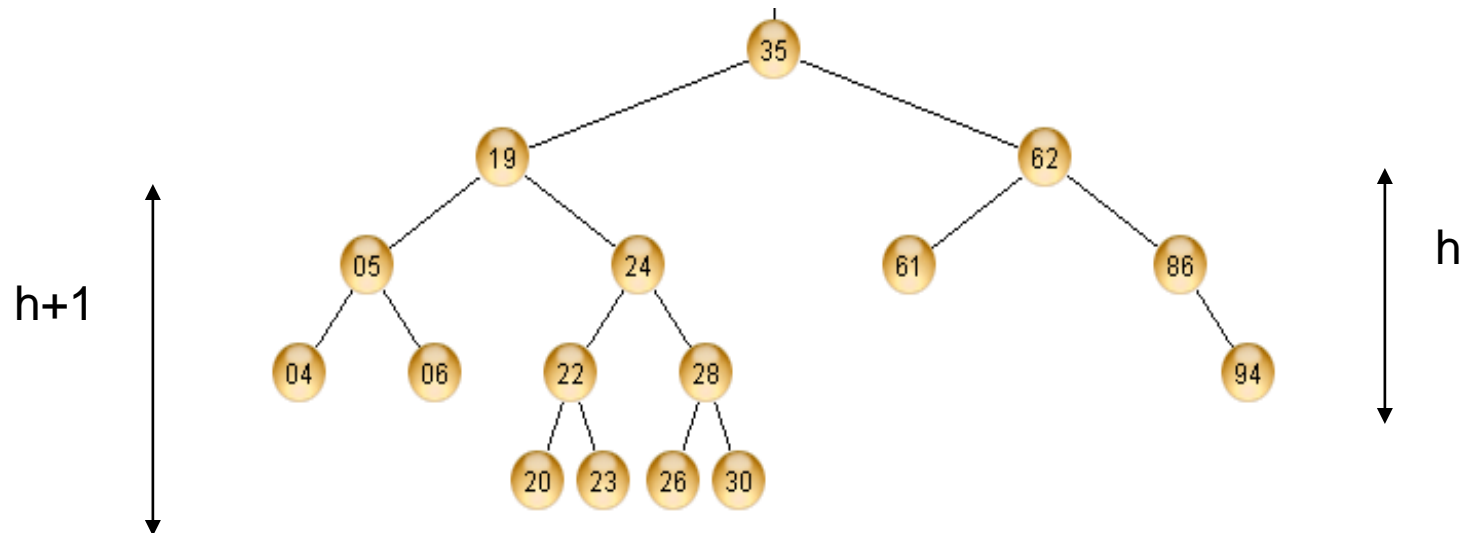
Περίπτωση 3

Διπλή Περιστροφή



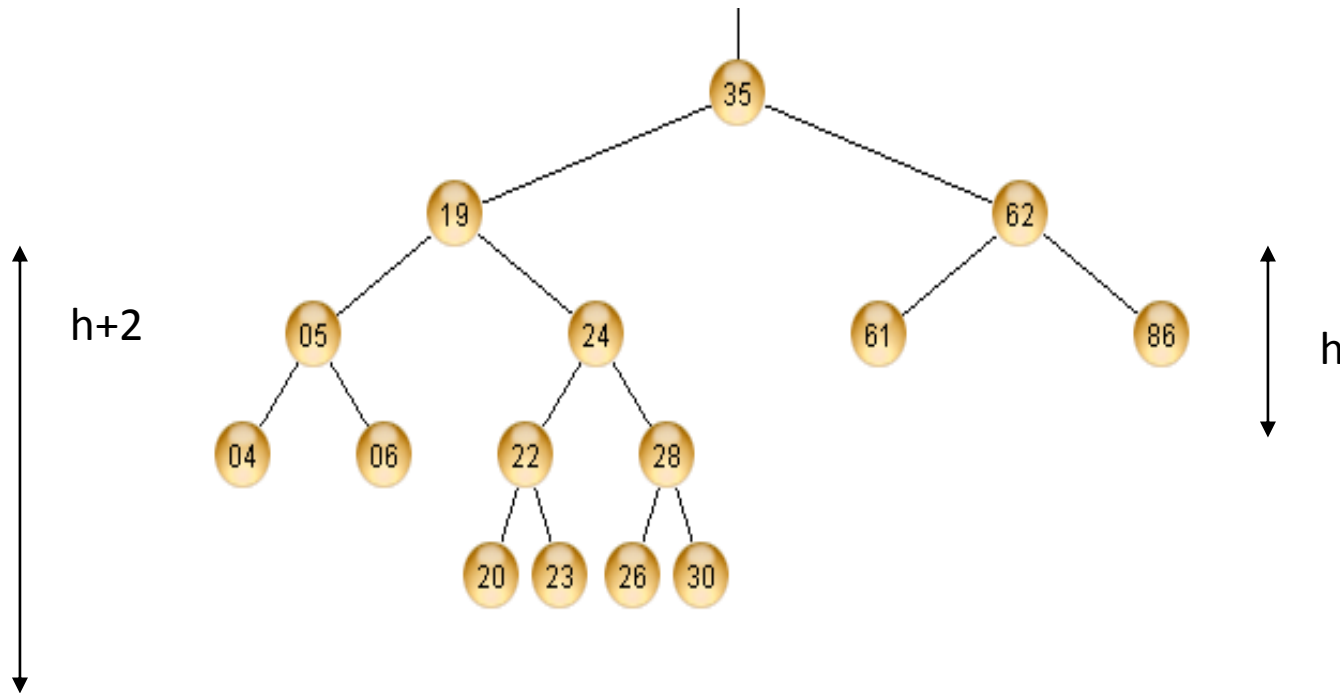
Περίπτωση 4

Διπλή Περιστροφή - Παράδειγμα



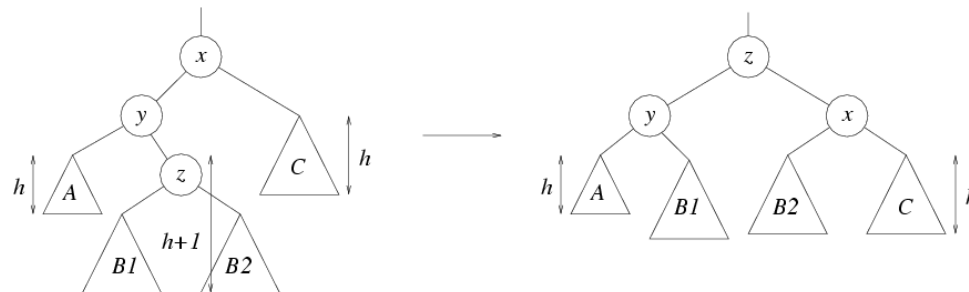
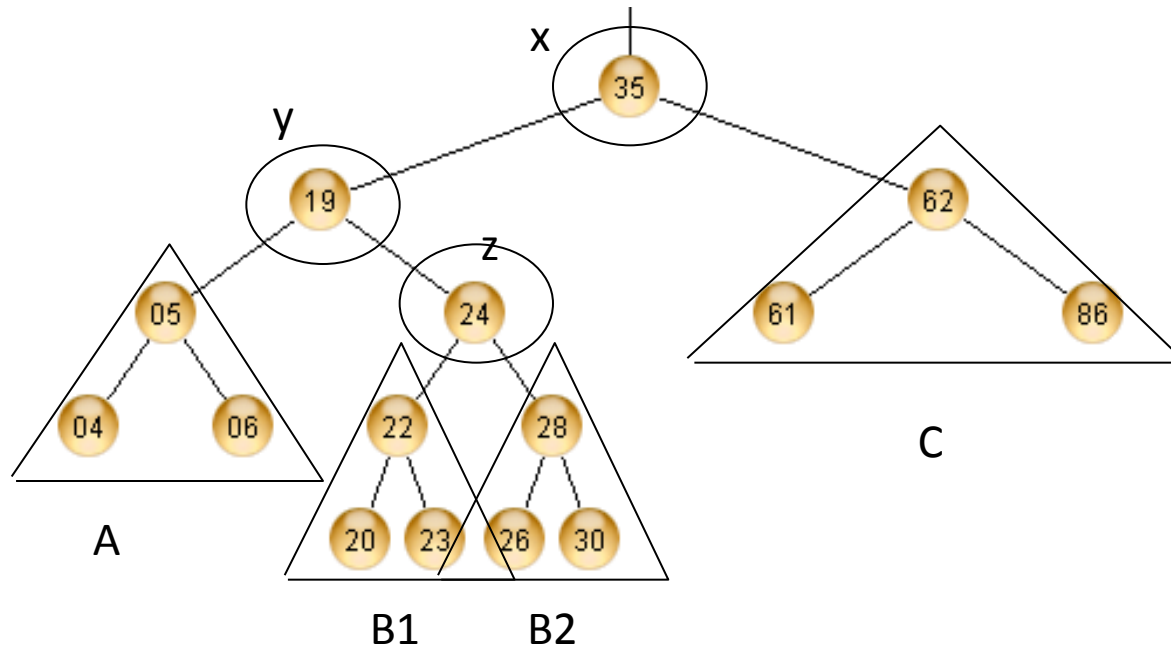
Διαγραφή του κόμβου 94

Διπλή Περιστροφή - Παράδειγμα

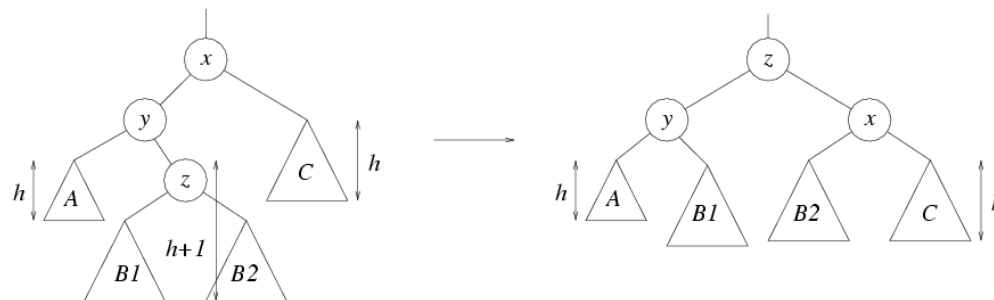
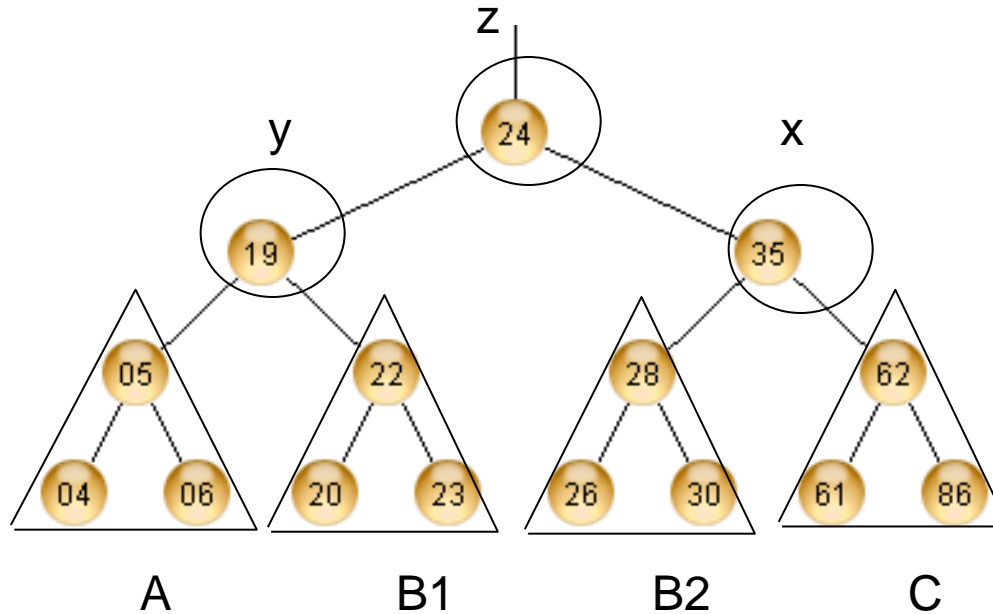


Παραβιάζεται ο ορισμός του AVL

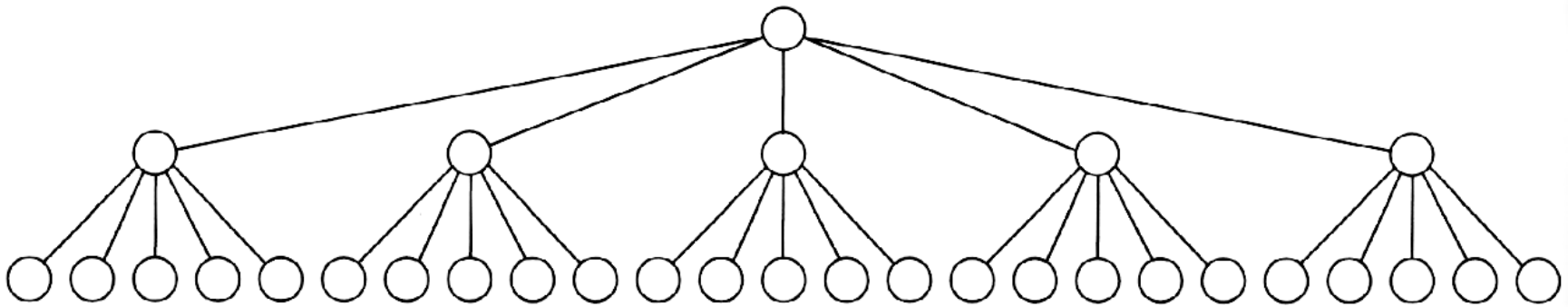
Διπλή Περιστροφή - Παράδειγμα



Μετά τη Διπλή Περιστροφή

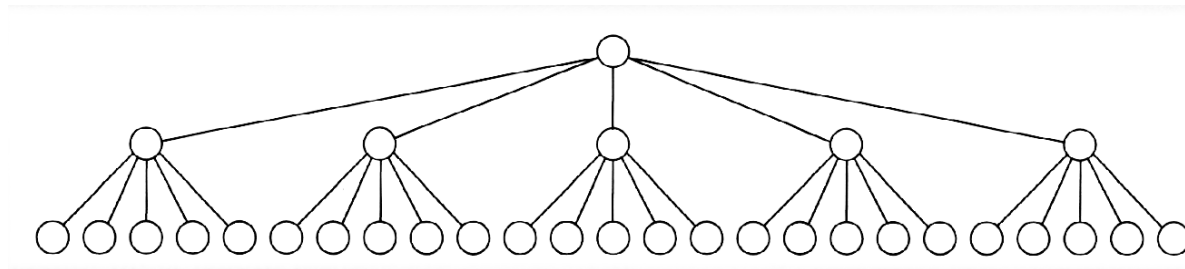


Δένδρα Πολλών Δρόμων

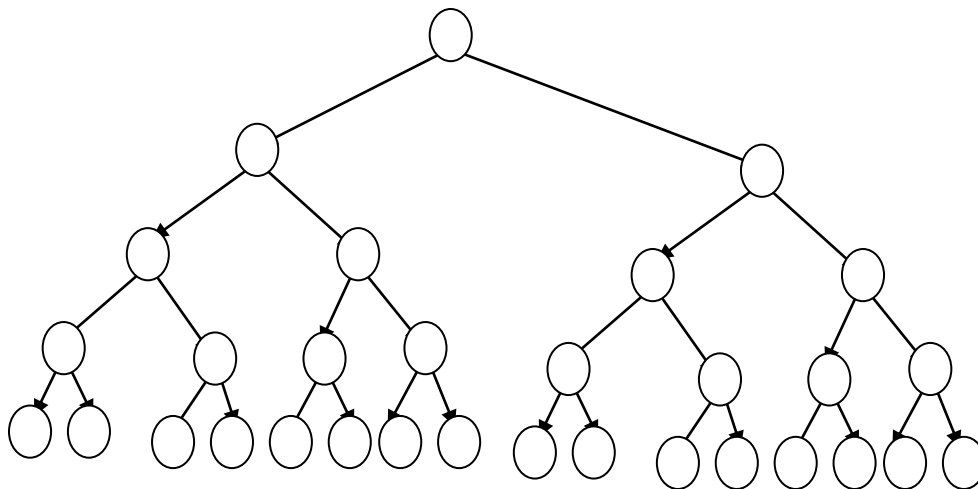


Δένδρο 5 δρόμων

Παράδειγμα



Δένδρο 5 δρόμων
31 κόμβοι
3 επίπεδα



Δυαδικό δένδρο
31 κόμβοι
5 επίπεδα



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Απόστολος Παπαδόπουλος. «Δομές Δεδομένων. Δυαδικά Δένδρα». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://eclass.auth.gr/courses/OCRS389/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>





Τέλος ενότητας

Επεξεργασία: <Μαυρίδης Απόστολος>
Θεσσαλονίκη, <Εαρινό εξάμηνο 2013-2014>



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Σημειώματα

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

