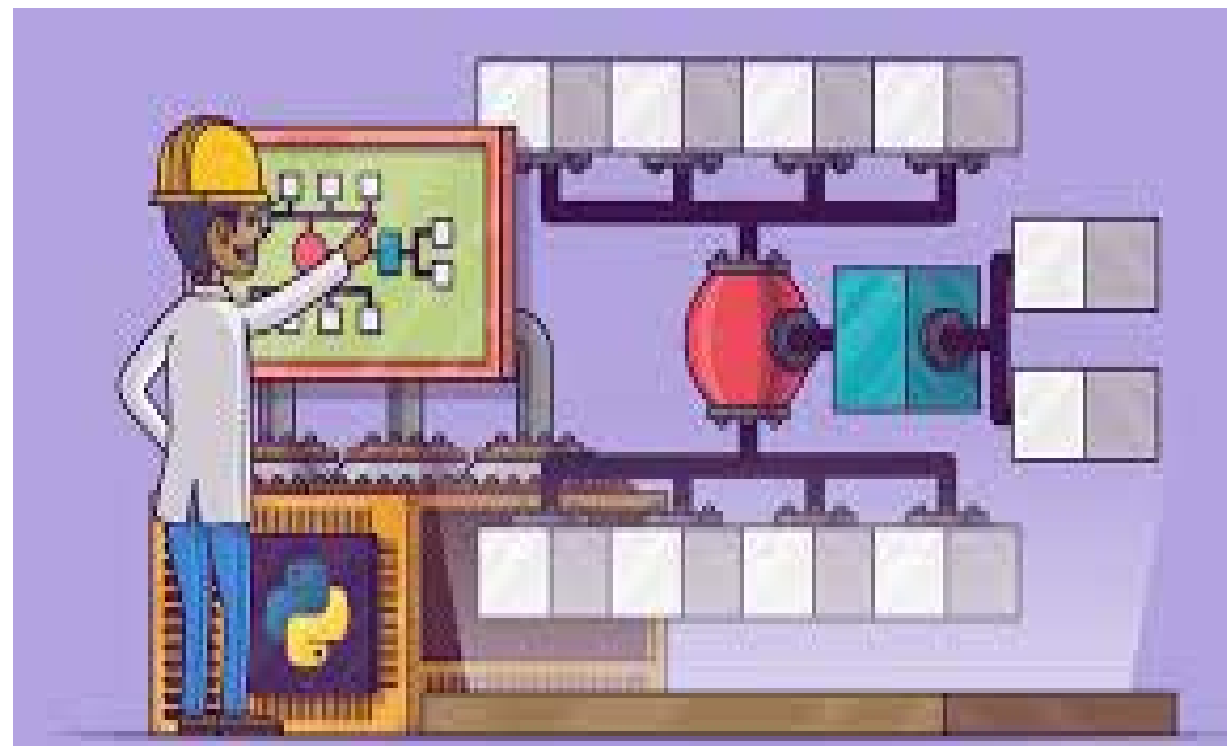




ساختمان داده ها

مدرس:
سمانه حسینی سمنانی

دانشگاه صنعتی اصفهان - دانشکده برق و
کامپیوتر





درخت ها

- مفاهیم اولیه
- پیمایش درخت
- درخت دودویی معادل



درخت ها

مفاهیم اولیه



مفاهیم اولیه

- درخت آزاد
- درخت ریشه دار
- اولاد یک گره
- اجداد یک گره
- جنگل
- رابطه پدر و فرزندی
- زیر درخت ریشه دار
- درجه هر نود



مفاهیم اولیه

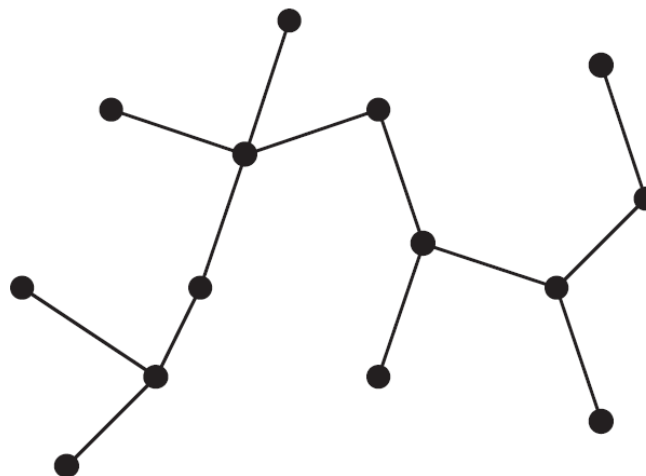
- عمق یک گره
- ارتفاع نود V
- ارتفاع درخت
- درخت متوازن
- درخت کاملاً متوازن
- درخت k تایی
- درخت مرتب



درخت ها

• درخت یا درخت آزاد

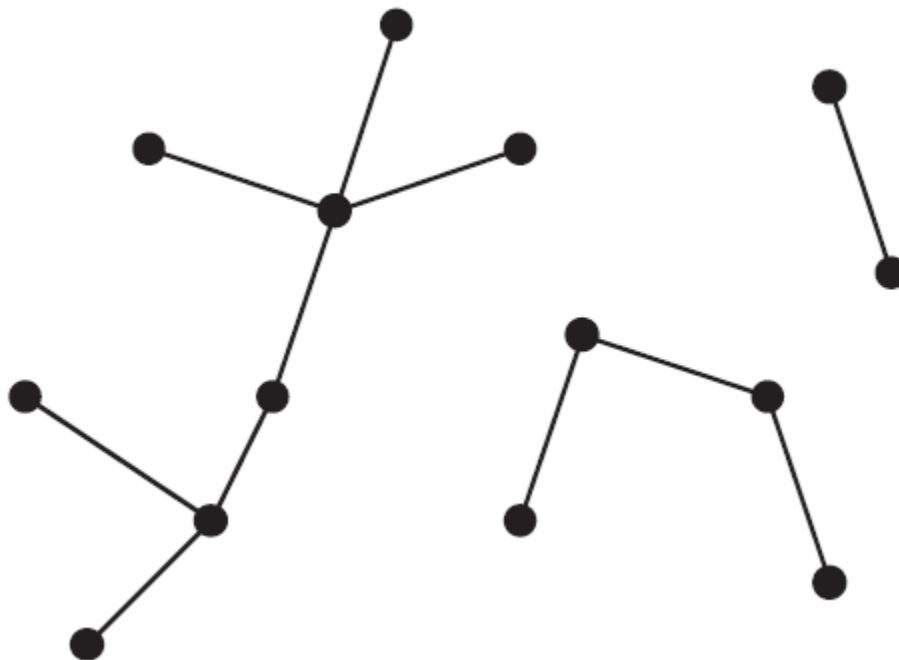
- A free tree is a connected, acyclic, undirected graph.





جنگل

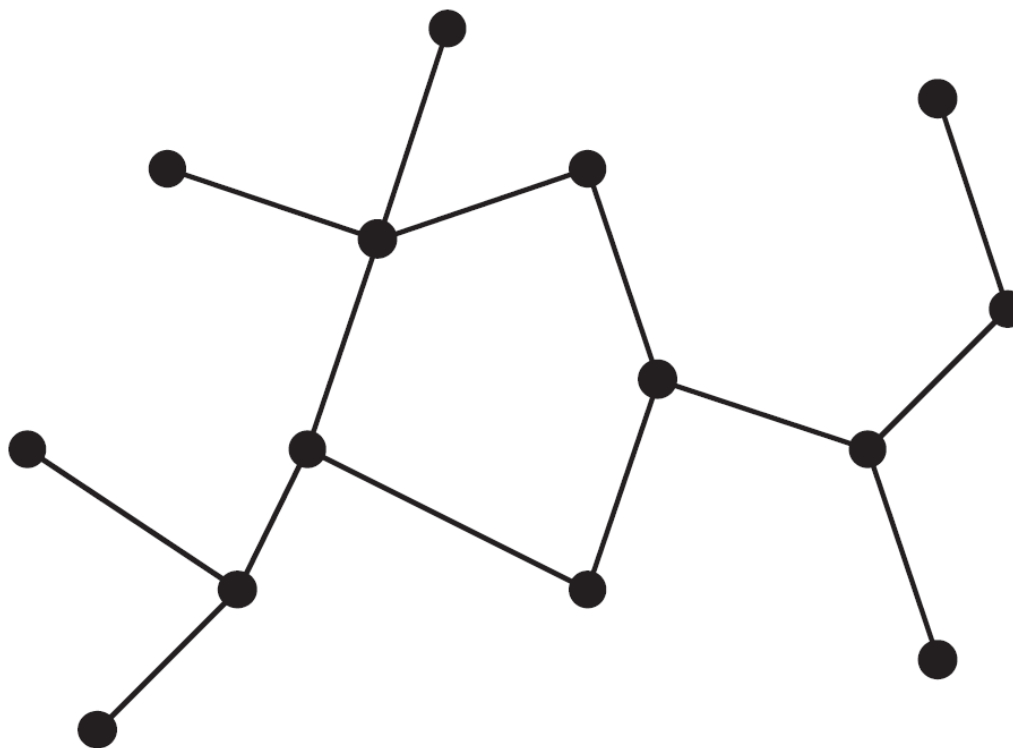
- If an undirected graph is acyclic but possibly disconnected, it is a forest.





درخت ها

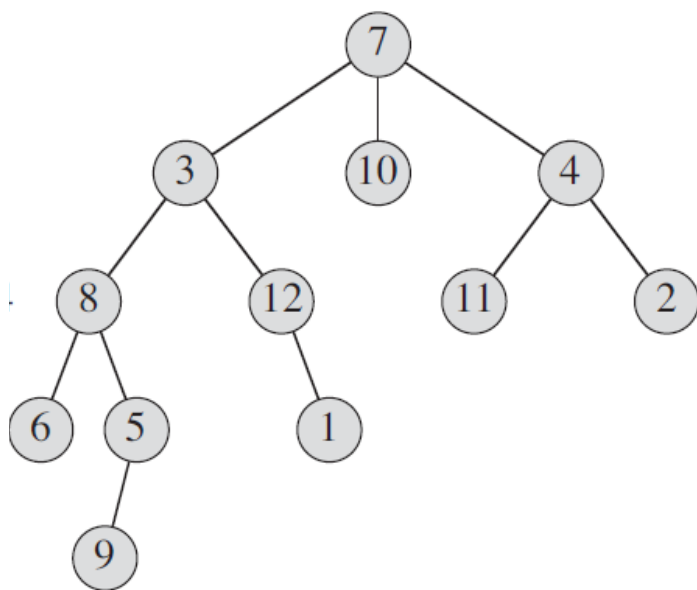
- Tree or forest?





درخت ریشه دار

- A **rooted tree** is a free tree in which one of the vertices is distinguished from the others (root).
- We often refer to a vertex of a rooted tree as a node of the tree.

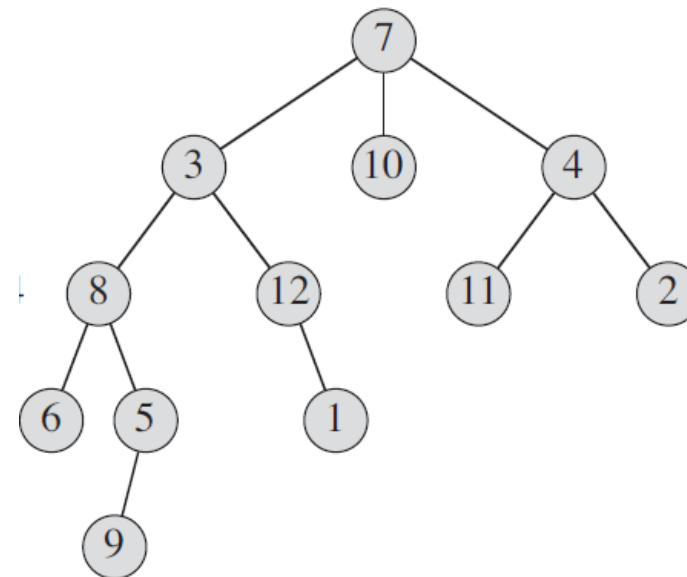


- یال ها جهت دهی می شوند.
- از بالا به پایین در شکل روبرو.
- درجه ریشه = صفر
- درجه ورودی هر نود غیر از ریشه = یک
- از اینجا به بعد وقتی میگویم درخت، منظور درخت ریشه دار است.



رابطه پدر و فرزندی

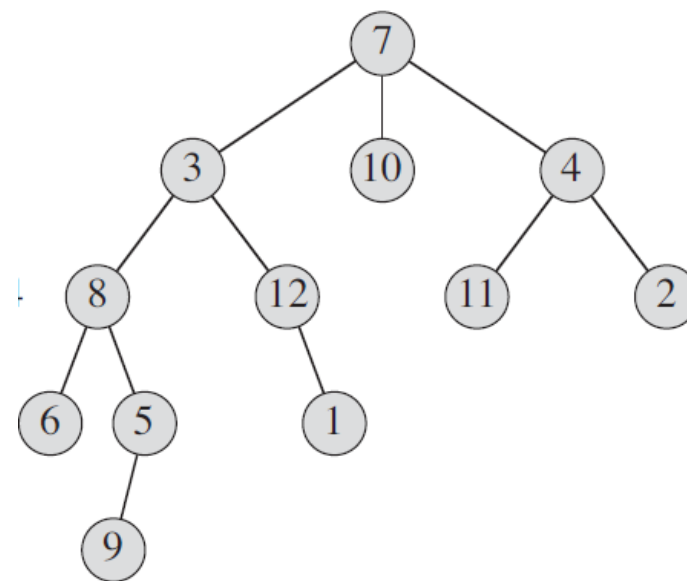
- Consider a node x in a rooted tree T with root r .
- We call any node y on the unique simple path from r to x an **ancestor** of x .
- If y is an ancestor of x , then x is a **descendant** of y .
- Every node is both an ancestor and a descendant of itself.
- If y is an ancestor of x and $x \neq y$, then y is a **proper ancestor** of x and x is a **proper descendant** of y .





رابطه پدر و فرزندی

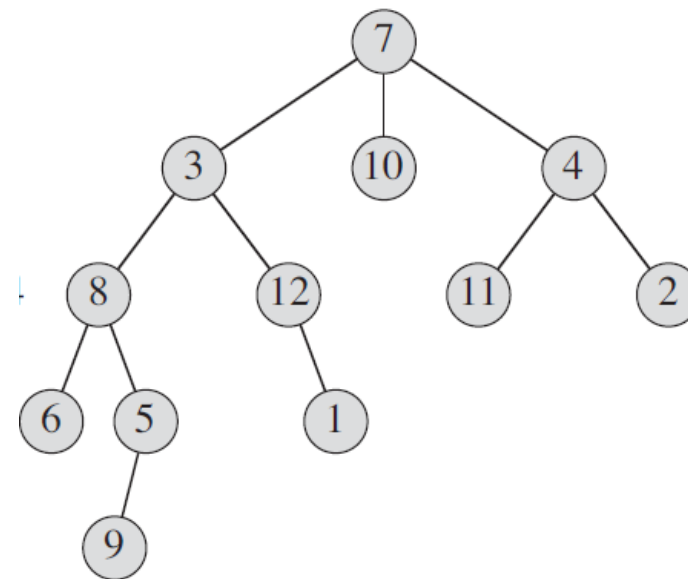
- If the last edge on the simple path from the root r of a tree T to a node x is (y,x) , then y is the **parent** of x , and x is a **child** of y .
- The root is the only node in T with no parent.
- If two nodes have the same parent, they are **siblings**.
- A node with no children is a **leaf** or **external node**
- A non leaf node is an **internal node**.





زیر درخت ریشه دار

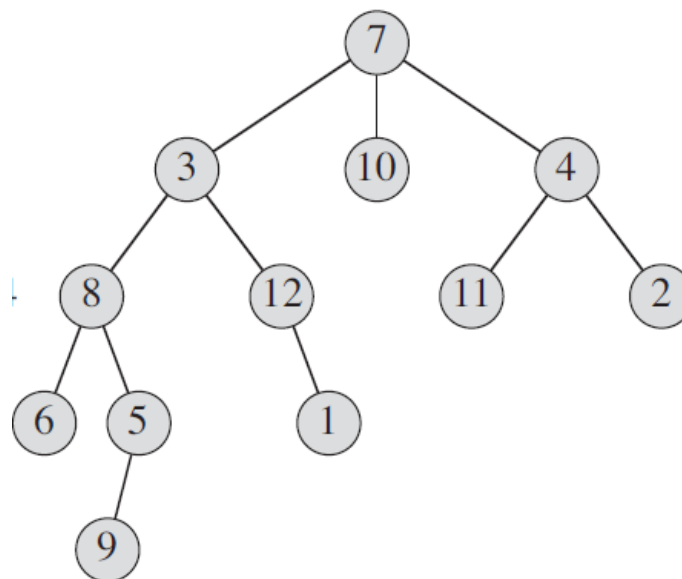
- The *subtree rooted at* x is the tree induced by descendants of x , rooted at x .
- For example, the subtree rooted at node 8 contains nodes 8, 6, 5, and 9.





درجه هر نود

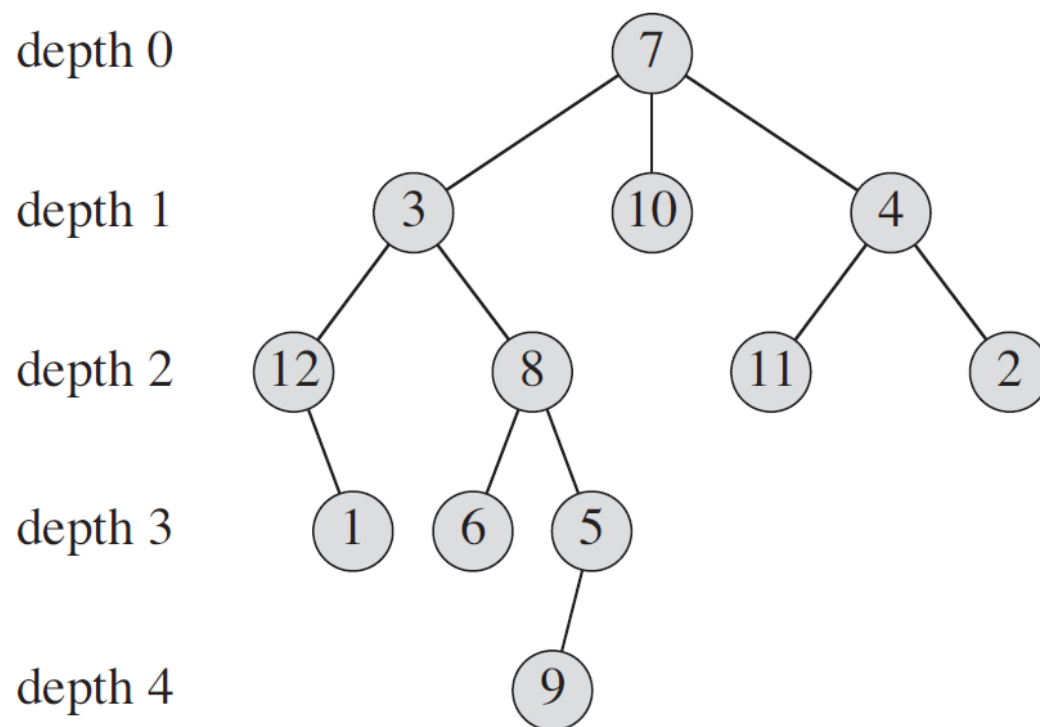
- The number of children of a node x in a rooted tree T equals the *degree* of x .





عمق هر نود

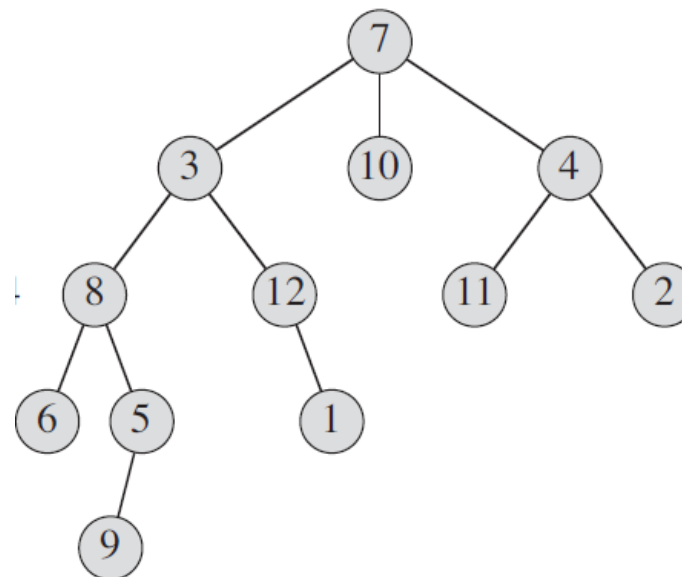
- The length of the simple path from the root r to a node x is the **depth** of x in T .





ارتفاع هر نود

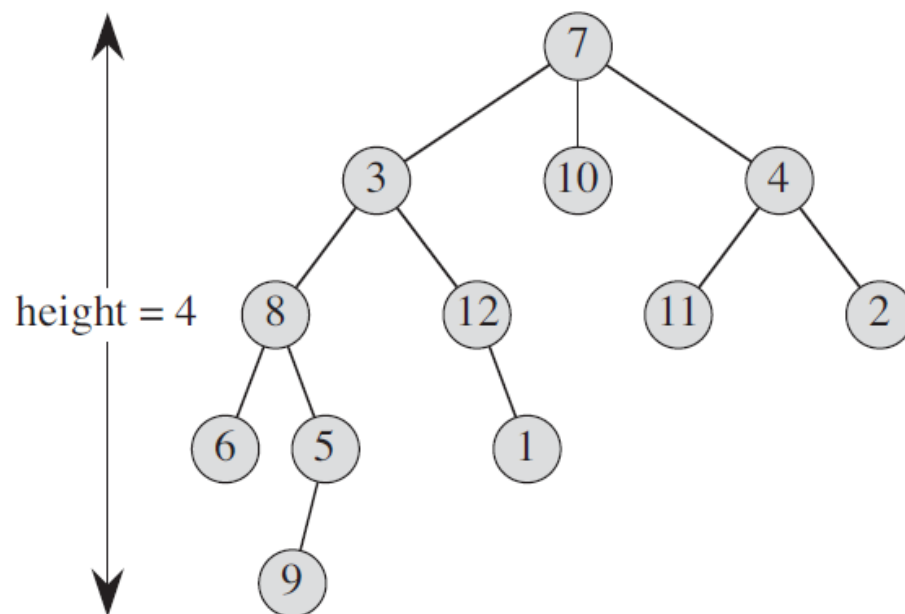
- The **height** of a node in a tree is the number of edges on the longest simple downward path from the node to a leaf, and the height of a tree is the height of its root.





ارتفاع درخت

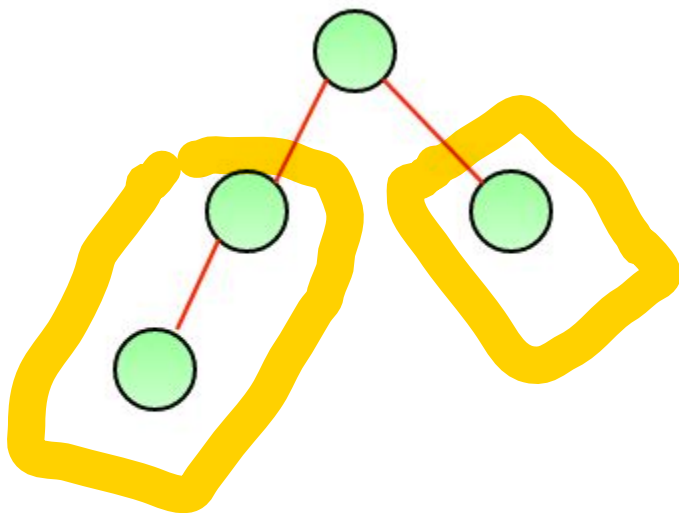
- **Height** of a tree is the height of its root.
- The height of a tree is also equal to the largest depth of any node in the tree.



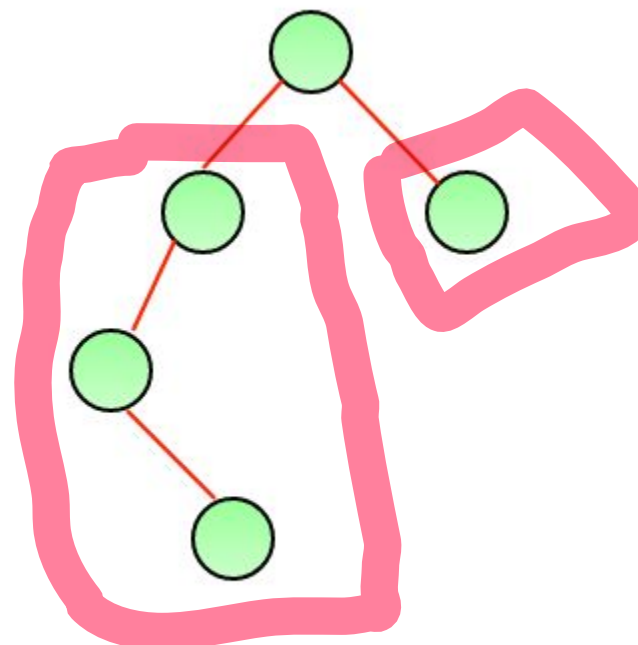


درخت متوازن

- A **balanced tree** is a tree where The left and right subtrees' heights differ by at most one.



A height balanced tree

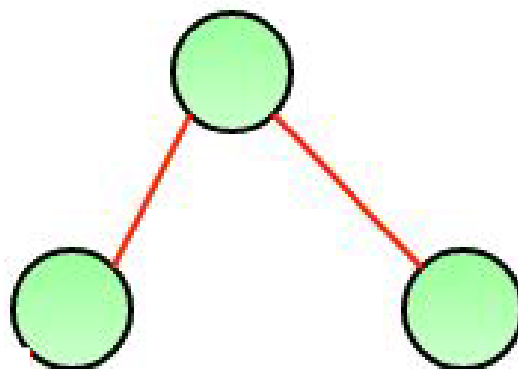


Not a height balanced tree



درخت کاملاً متوازن

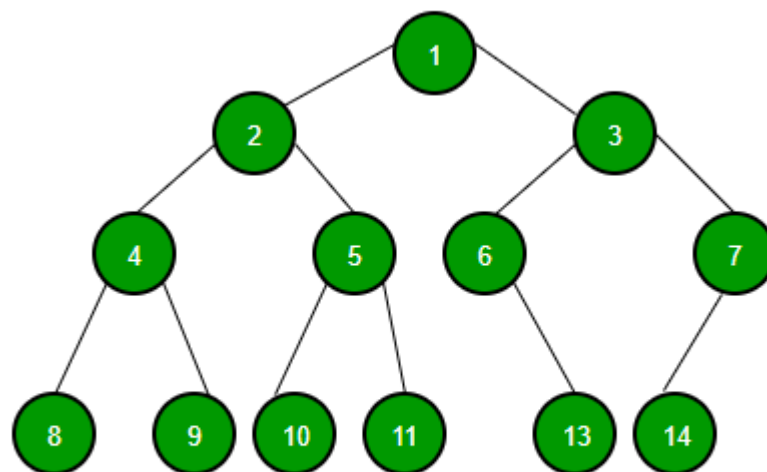
- A *complete balanced tree* is a tree where The left and right subtrees' heights differ by at most zero.





درخت k تایی

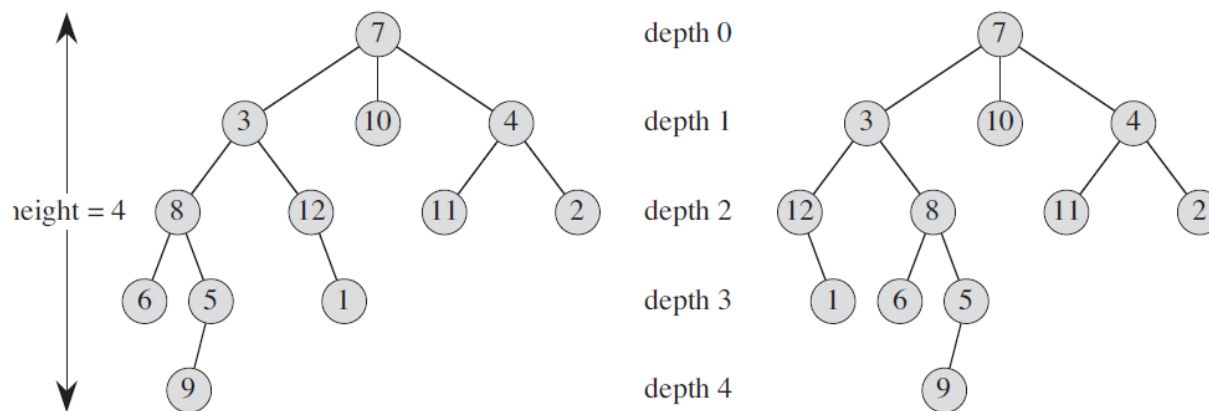
- درخت k تایی درختی است که در آن هر گره حداکثر k گره فرزند دارد.





درخت مرتب

- An **ordered tree** is a rooted tree in which the **children** of each node are ordered.
- That is, if a node has k children, then there is a first child, a second child, \dots , and a k_{th} child.



As a rooted tree, it is identical to the tree in (a), but as an ordered tree it is different, since the children of node 3 appear in a different order.



درخت ها

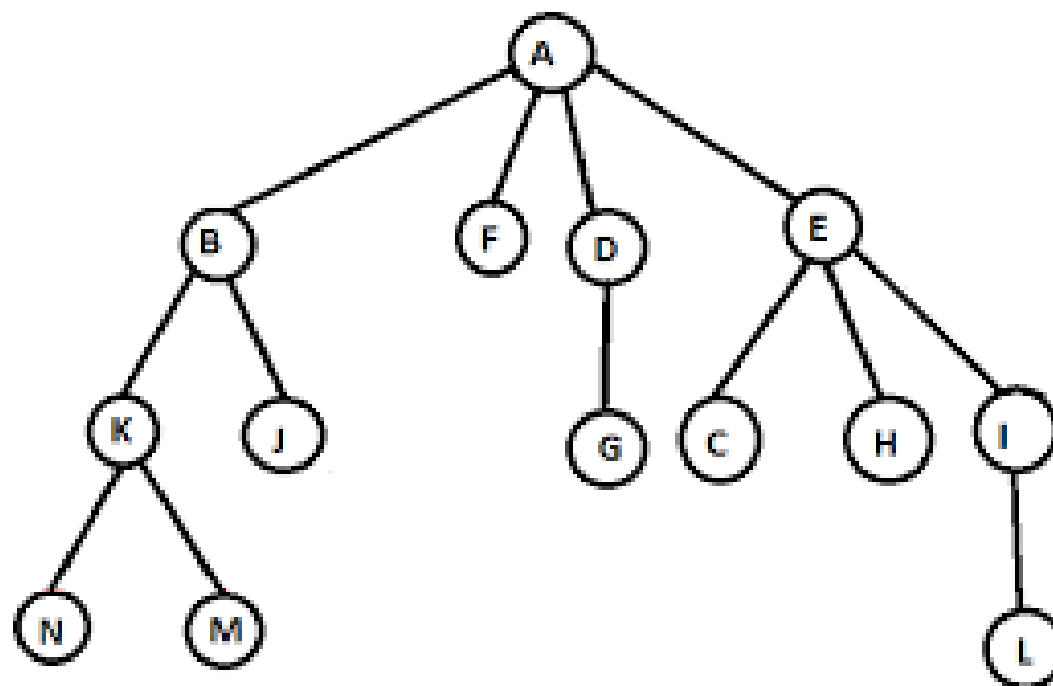
پیمایش درخت

Tree traversal



روش های مختلف پیمایش درخت

- Random
- Preorder
- Inorder
- Postorder
- **Mixed**





روش های مختلف پیمایش درخت

- Preorder

$$r, T_1 T_2 \cdots T_k$$

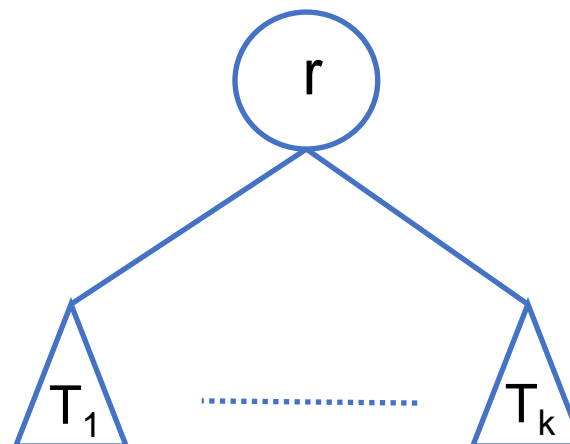
- Inorder

$$T_1, r, T_2 \cdots T_k$$

- Postorder

$$T_1 \cdots T_k, r$$

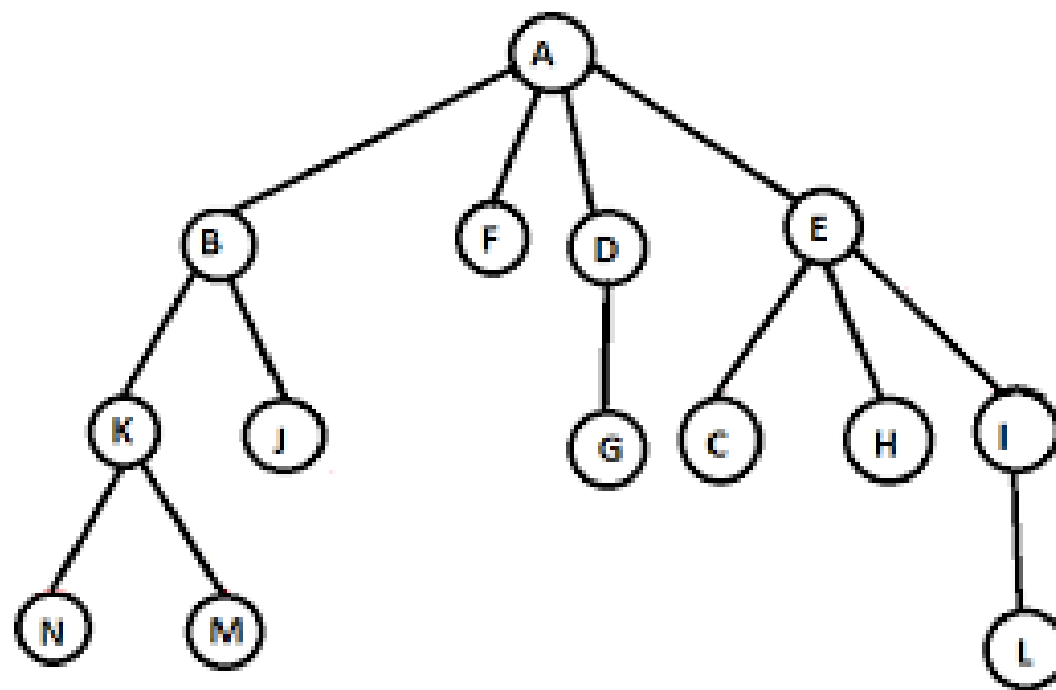
بازگشتی





مثال

- Preorder: A, B, K, N, M, J, F, D, G, E, C, H, I, L
- Inorder: N, K, M, B, J, A, F, G, D, C, E, H, L, I
- Postorder: N, M, K, J, B, F, G, D, C, H, L, I, E, A

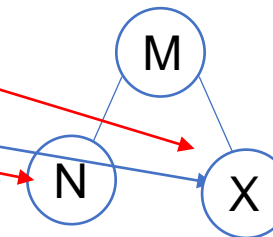




مثال

درخت دودویی

- Preorder: M, N, H, C, R, S, K, W, T, G, D, X, I, Y, A, J, P, O, E, Z, V, B, U, L, Q, F
- Postorder: C, W, T, K, S, G, R, H, D, N, A, O, E, P, J, Y, Z, I, B, Q, L, F, U, V, X, M
- Inorder?



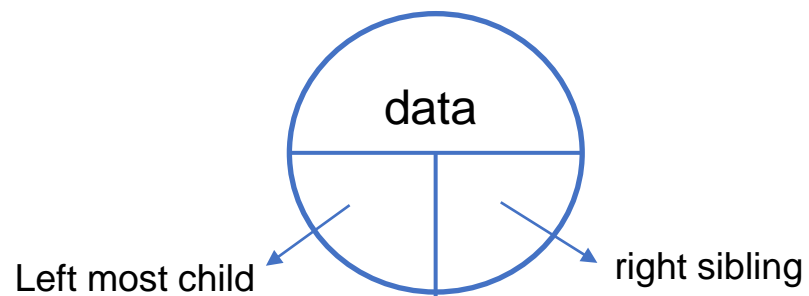
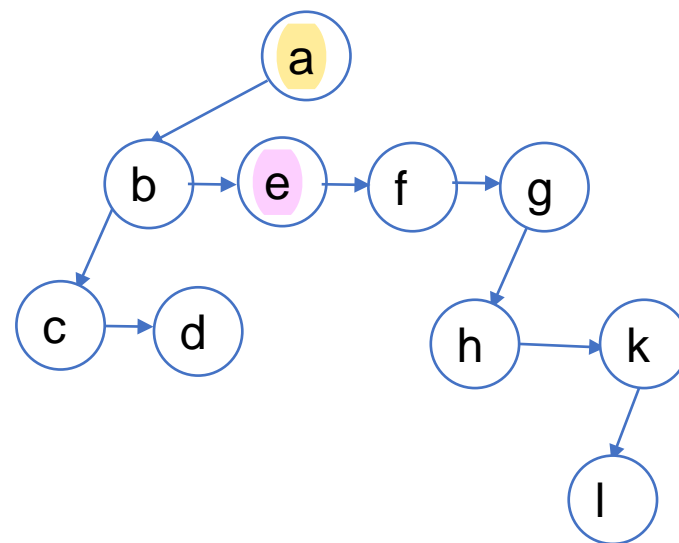
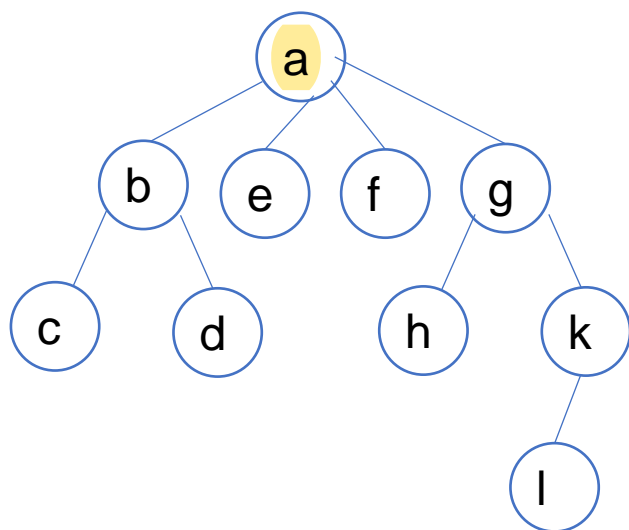


درخت ها

درخت دودویی معادل

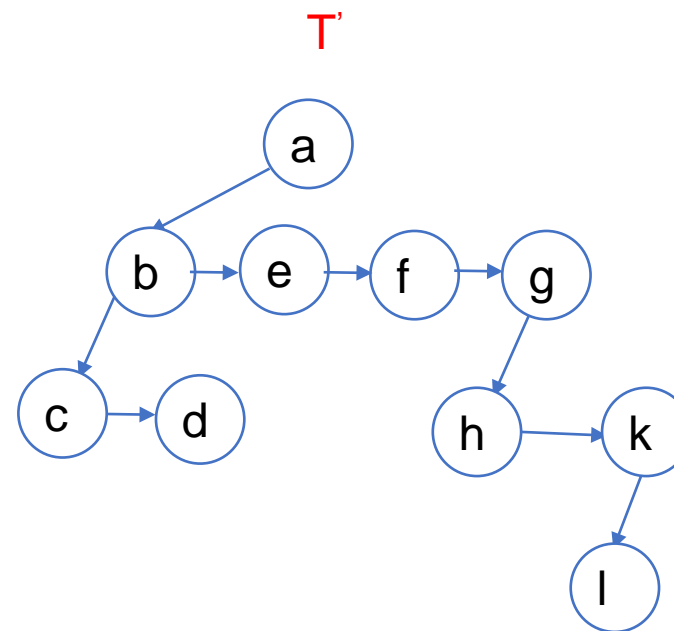
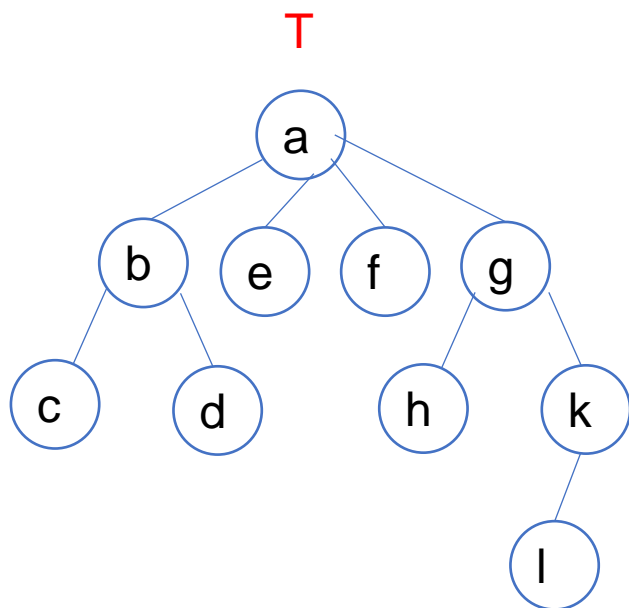


درخت دودویی معادل





درخت دودویی معادل



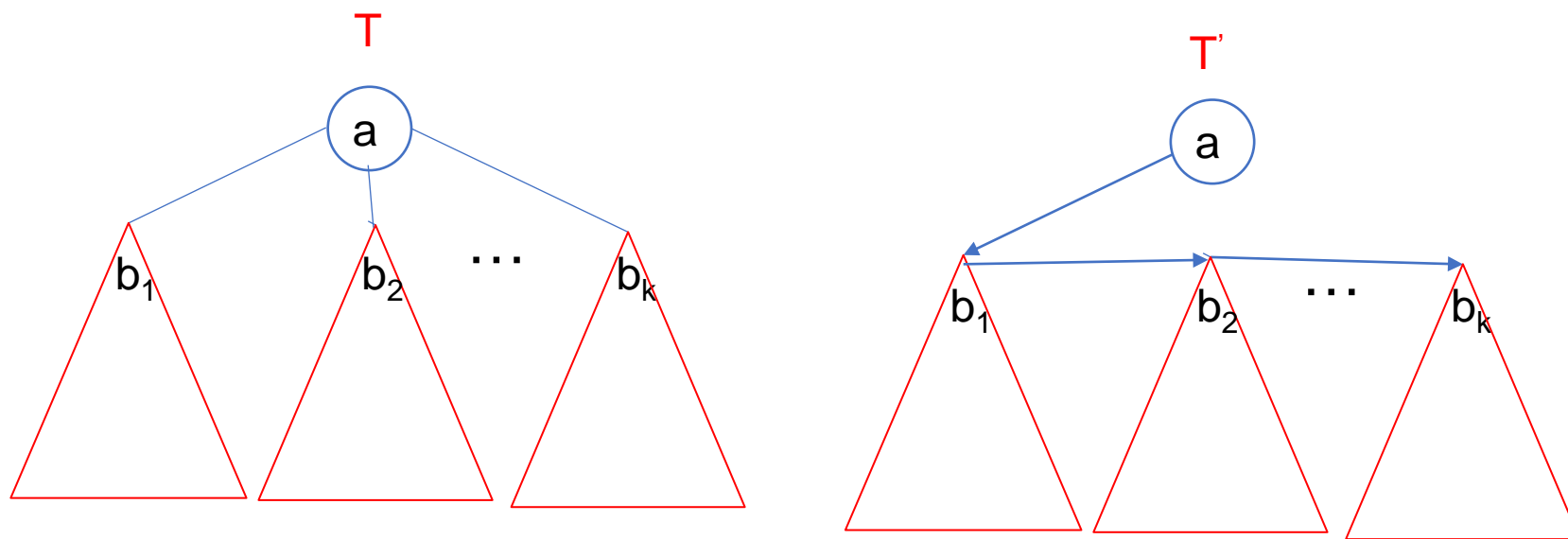
$preorder(T) = a, b, c, d, e, f, g, h, k, l$

$preorder(T') = a, b, c, d, e, f, g, h, k, l$

- چرا برابر شد؟
- آیا در حالت کلی این طور است؟



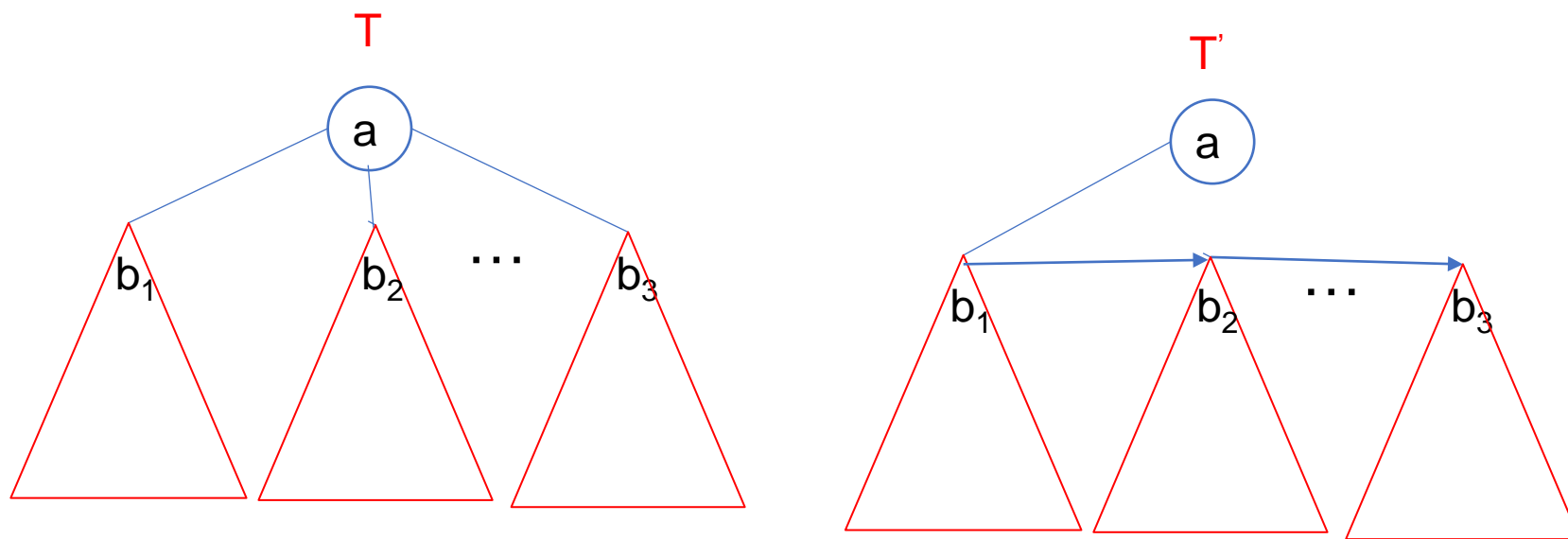
درخت دودویی معادل



- چرا برابر شد؟
- آیا در حالت کلی این طور است؟



درخت دودویی معادل



$$postorder(T') = postorder(T)$$

$$inorder(T) = inorder(T')$$