Veri Yapıları ve Algoritmalar

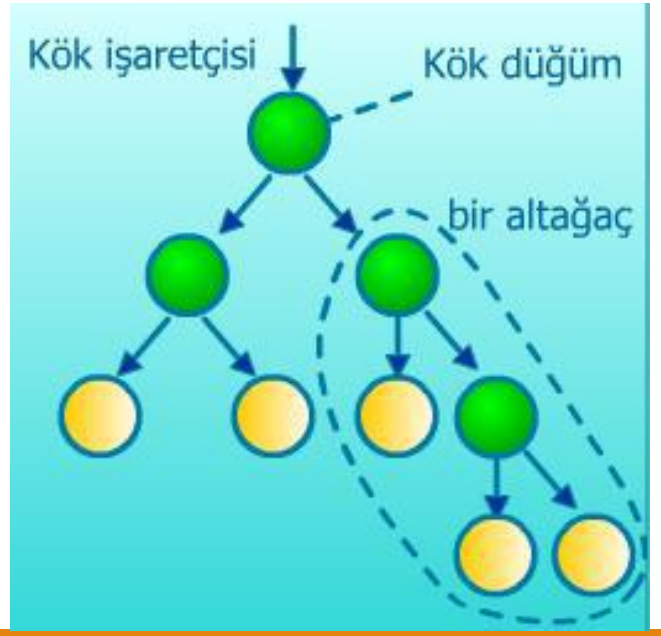
DR. ÖGR. ÜYESİ MEHMET AKİF BÜLBÜL

2023-2024 GÜZ YARIYILI

AĞAÇ VERİ MODELİ

Temel Kavramlar

Bağlı listeler, yığıtlar doğrusal (linear) veri yapılarıdır. Ağaçlar ise doğrusal olmayan belirli niteliklere sahip iki boyutlu veri yapılarıdır.



Ağaçlar hiyerarşik ilişkileri göstermek için kullanılır.

Her ağaç node'lar ve kenarlardan (edge) oluşur.

Herbir node(düğüm) bir nesneyi gösterir.

Herbir kenar (bağlantı) iki node arasındaki ilişkiyi gösterir.

Temel Kavramlar

Ağaçlardaki düğümlerden iki veya daha fazla bağ çıkabilir. İkili ağaçlar (binary trees), düğümlerinde en fazla iki bağ içeren ağaçlardır. Ağacın en üstteki düğümüne kök (root) adı verilir. Uygulamaları:

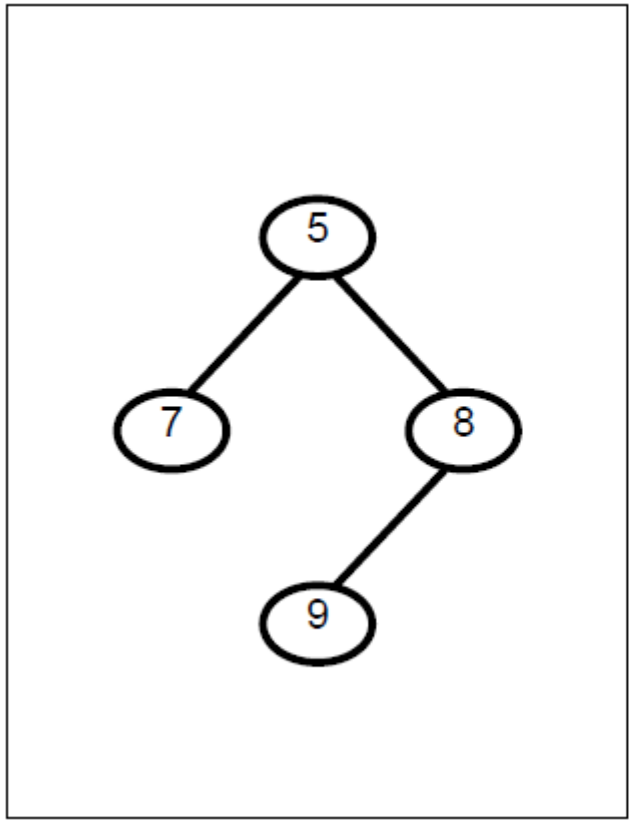
Organizasyon şeması

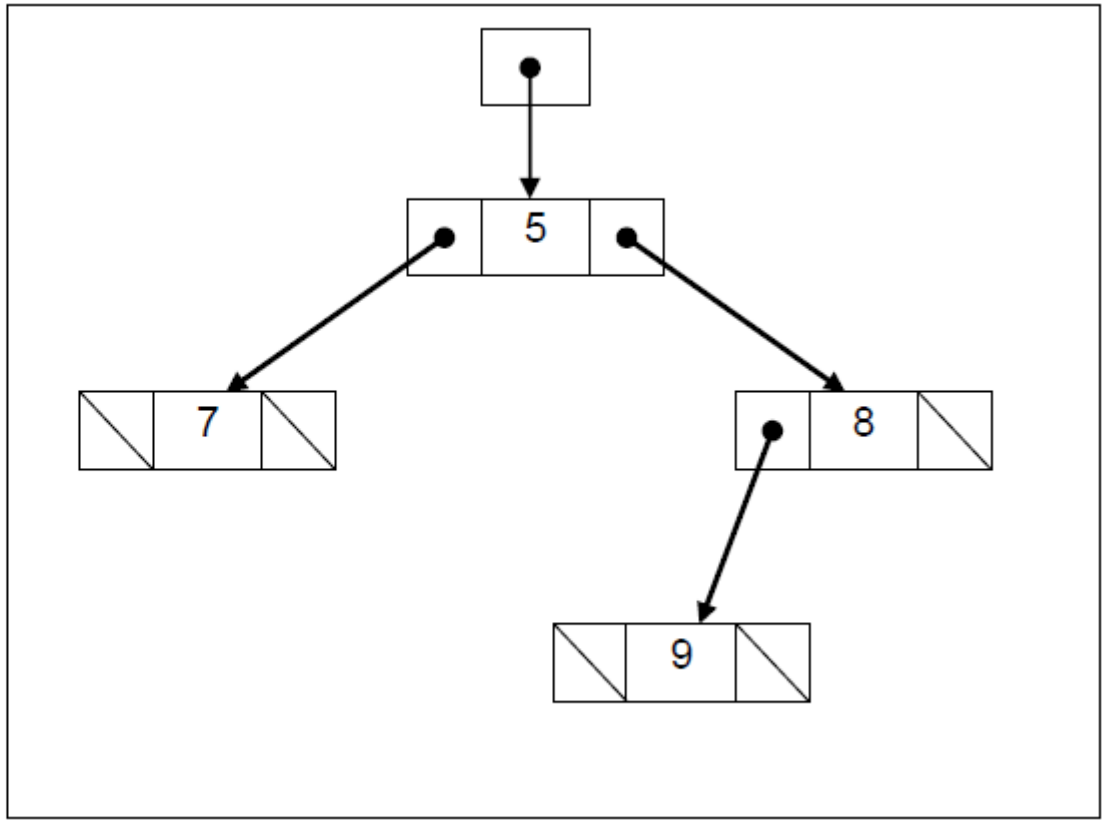
Dosya sistemleri

Programlama ortamları

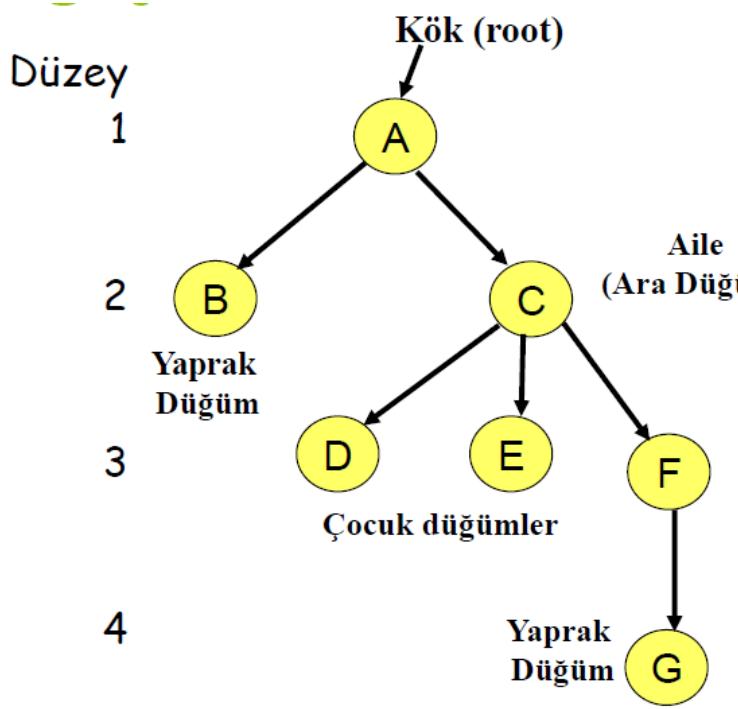
Temel Kavramlar

İkili Ağacın Grafiksel Gösterimleri





Temel Kavramlar



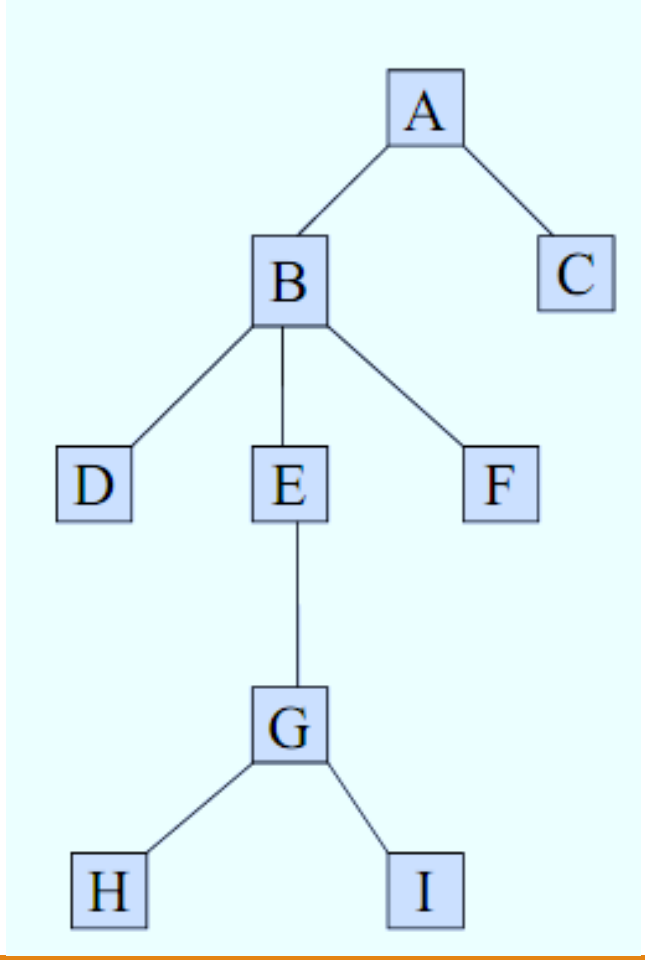
- **Şekil :** Ağaçlarda düzeyler

Şekildeki ağaç, A düğümü kök olmak üzere 7 düğümden oluşmaktadır.

Sol alt ağaç B kökü ile başlamakta ve sağ alt ağaç da C kökü ile başlamaktadır.

A'dan solda B'ye giden ve sağda C'ye giden iki dal (branch) çıkmaktadır.

Temel Kavramlar



Ağaç Türleri

En çok bilinen ağaç türleri ikili arama ağacı olup kodlama ağacı, sözlük ağacı, kümeleme ağacı gibi birçok ağaç uygulaması vardır.

İkili Arama Ağacı (Binary Search Tree):

İkili arama ağacında bir düğüm en fazla iki tane çocuğa sahip olabilir ve alt/çocuk bağlantıları belirli bir sırada yapılır.

Kodlama Ağacı (Coding Tree):

Bir kümedeki karakterlere kod ataması için kurulan ağaç şeklidir. Bu tür ağaçlarda kökten başlayıp yapraklara kadar olan yol üzerindeki bağlantı değerleri kodu verir.

Sözlük Ağacı(Dictonary Tree):

Bir sözlükte bulunan sözcüklerin tutulması için kurulan bir ağaç şeklidir. Amaç arama işlemini en performanslı bir şekilde yapılması ve belleğin optimum kullanılmasıdır.

Kümeleme Ağacı (Heap Tree):

Bir çeşit sıralama ağacıdır. Çocuk düğümler her zaman aile düğümlerinden daha küçük değerlere sahip olur.

İkili Ağaç (Binary Tree)

Sonlu düğümler kümesidir. Bu küme boş bir küme olabilir. Boş değilse şu kurallara uyar.

Kök olarak adlandırılan özel bir düğüm vardır.

Her düğüm en fazla iki düğüme bağlıdır.

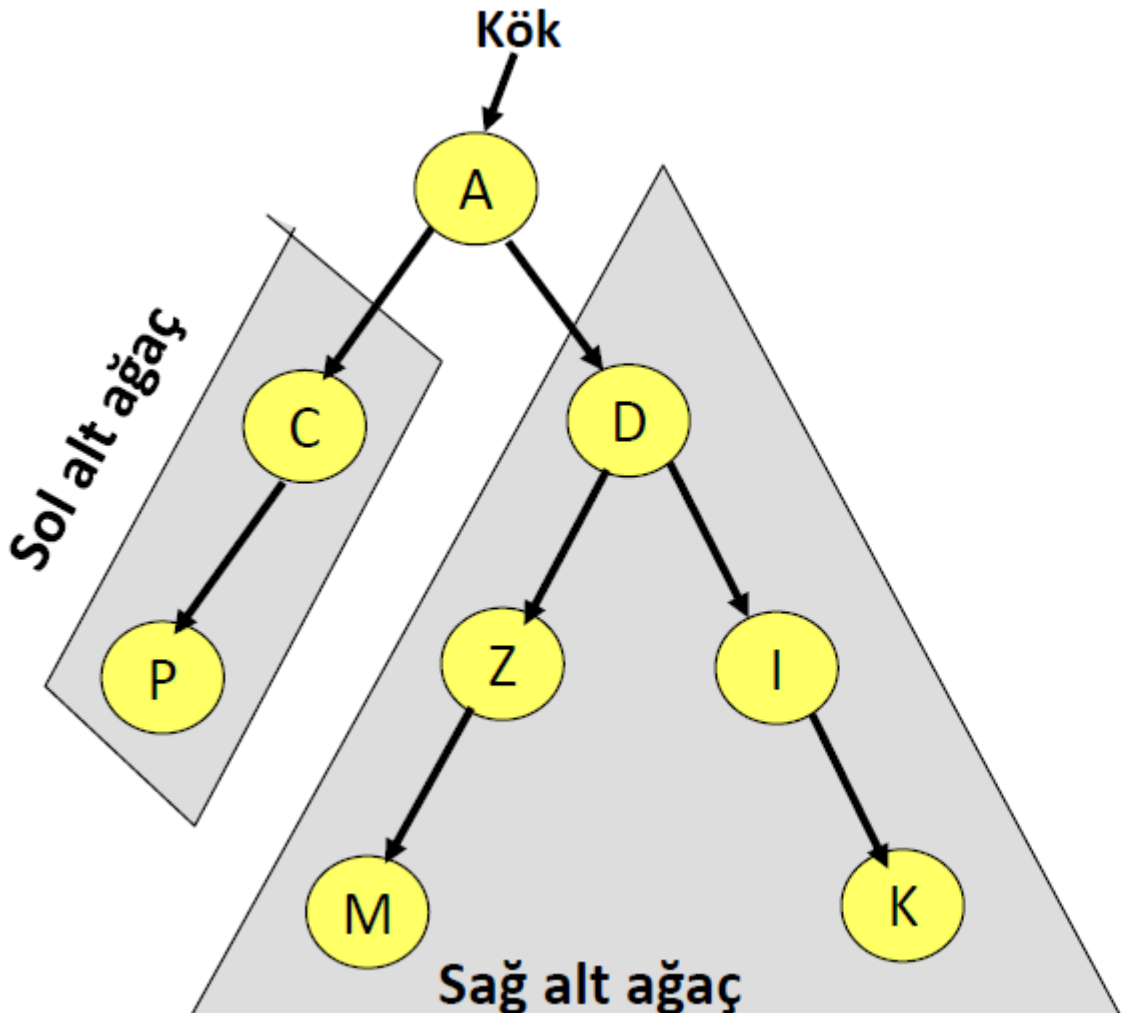
Left child : Bir node'un sol işaretçisine bağlıdır.

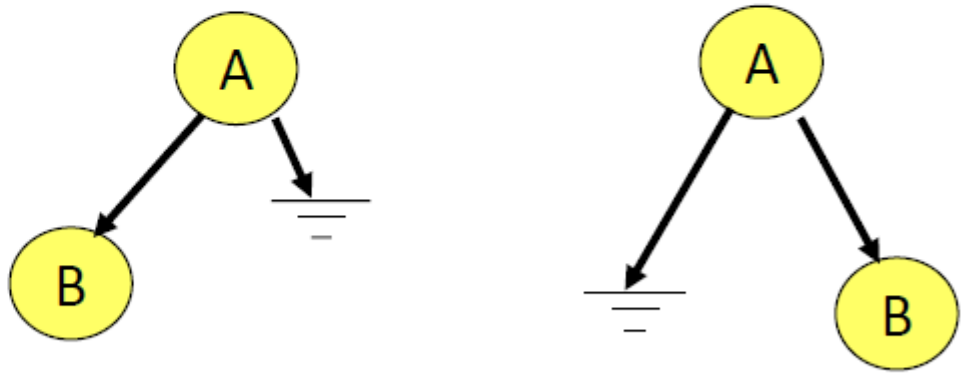
Right child : Bir node'un sağ işaretçisine bağlıdır.

Kök hariç her düğüm bir daldan gelmektedir.

Tüm düğümlerden yukarı doğru çıkıldıkça sonuçta köke ulaşılır.

İkili Ağaç (Binary Tree)

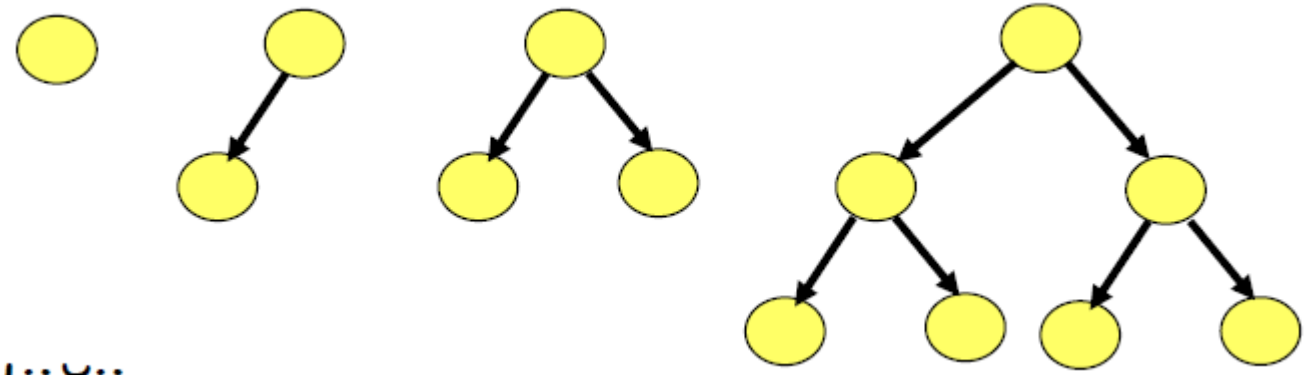




İki farklı ikili ağaç

İkili Ağaç (Binary Tree)

N tane düğüm veriliyor, İkili ağacın minimum derinliği nedir.



Derinlik 1: N = 1, 1 düğüm $2^1 - 1$

Derinlik 2: N = 2, 3 düğüm, $2^2 - 1$ düğüm

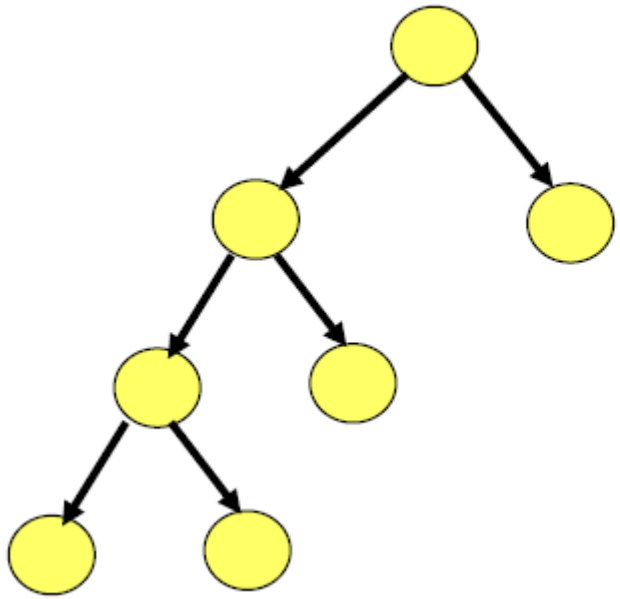
Herhangi bir d derinliğinde, N = ?

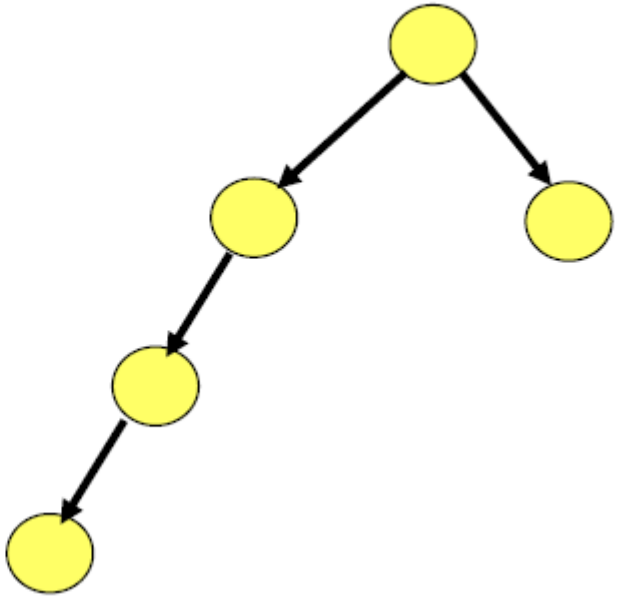
Derinlik d: N = $2^d - 1$ düğüm (tam bir ikili ağaç)

En küçük derinlik: **$\Theta(\log N)$**

Proper (düzgün) Binary Trees

Yaprak olmayan düğümlerin tümünün iki çocuğu olan T ağacı proper(düzgün) binary tree olarak adlandırılır.





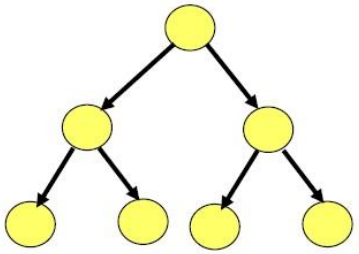
Full Binary Tree

Full binary tree:
1- Her yaprağı aynı derinlikte olan

2- Yaprak olmayan düğümlerin tümünün iki çocuğu olan ağaç Full (Strictly) Binary Tree'dir.
Bir full binary tree'de n tane yaprak varsa bu ağaçta toplam 2n-1 düğüm vardır. Başka bir şekilde ifade edilirse,

Eğer T ağacı boş ise, T yüksekliği 0 olan bir full binary ağaçtır.
T ağacının yüksekliği h ise ve yüksekliği h'den küçük olan tüm node'lar iki child node'a sahipse, T full binary tree'dir.

Full binary tree'de her node aynı yüksekliğe eşit sağ ve sol altağaçlara sahiptir.



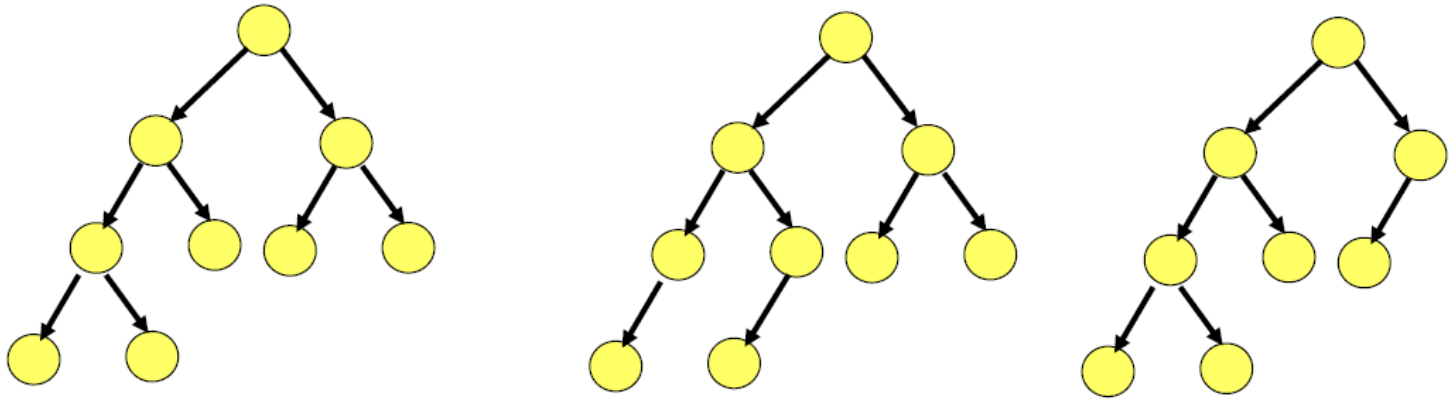
Complete Binary Tree

Full binary tree'de, yeni bir derinliğe soldan sağa doğru düğümler eklendiğinde oluşan ağaçlara Complete Binary Tree denilir.

Böyle bir ağaçta bazı yapraklar diğerlerinden daha derindir. Bu nedenle full binary tree olmayabilirler. En derin düzeyde düğümler olabildiğince soldadır.

T, n yükseklikte complete binary tree ise, tüm yaprak node'ları n veya n-1 derinliğindedir ve yeni bir derinliğe soldan sağa doğru ekleme başlanır.

Her node iki tane child node'a sahip olmayabilir.

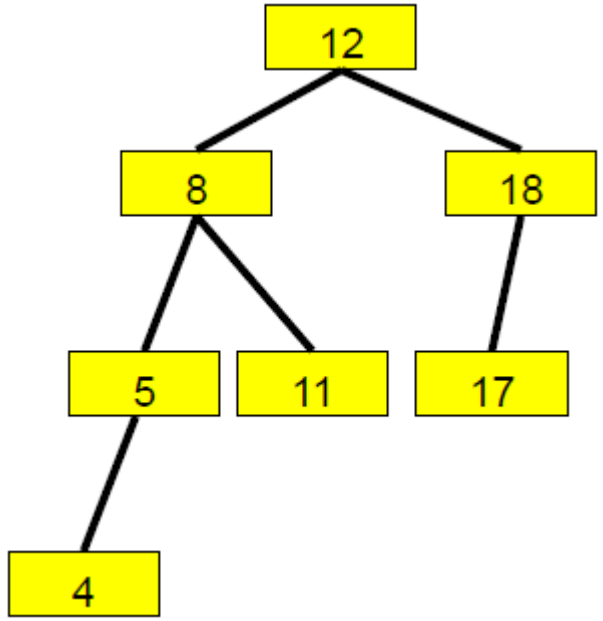


Balanced Binary Trees

Yüksekliği ayarlanmış ağaçlardır.
Bütün düğümler için sol altağacın yüksekliği ile sağ altağacın yüksekliği arasında en fazla bir fark varsa balanced binary tree olarak adlandırılır.

Complete binary tree'ler aynı zamanda balanced binary tree'dir.

Her balanced binary tree, complete binary tree olmayabilir. (Neden?)



Geçiş İşlemleri
İkili Ağaçlar Üzerindeki Geçiş İşlemleri

Geçiş (traverse), ağaç üzerindeki tüm düğümlere uğrayarak gerçekleştirilir. Ağaçlar üzerindeki geçiş işlemleri, ağaçtaki tüm bilgilerin listelenmesi veya başka amaçlarla yapılır.

Doğrusal veri yapılarında baştan sona doğru dolaşmak kolaydır. Ağaçlar ise düğümleri doğrusal olmayan veri yapılarıdır. Bu nedenle farklı algoritmalar uygulanır.

İkili Ağaçlar Üzerindeki Geçiş İşlemleri

Preorder (depth-first order) Dolaşma (Traversal) (Önce Kök)
Köke uğra (visit)
Sol alt ağacı preorder olarak dolaş.
Sağ alt ağacı preorder olarak dolaş.

Inorder (Symmetric order) Dolaşma(Ortada Kök)
Sol alt ağacı inorder'a göre dolaş
Köke uğra (visit)
Sağ alt ağacı inorder'a göre dolaş.

Postorder Dolaşma (Sonra Kök)
Sol alt ağacı postorder'a göre dolaş
Sağ alt ağacı postorder'a göre dolaş.
Köke uğra (visit)

Level order Dolaşma (Genişliğine dolaşma)
Köke uğra
Soldan sağa ağacı dolaş

İkili Ağaçlar Üzerindeki Geçiş İşlemleri