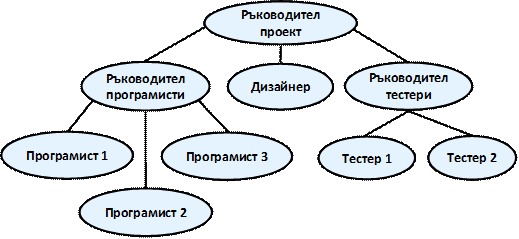
Видове дървета

# Въведение

Формалната теоретико-множествена дефиниция на дърво е свързан, ацикличен граф. Дърветата като структура от данни се ползват, защото естествено представят йерархия/наредба на някакви обекти. Още една причина за използването им е, че някои видове дървета имат много добра времева сложност за основните операции като търсене/добавяне/изтриване на елемент.

1 Пример за дървовидна структура:

ю

# Терминология

Някои основни понятия за дърветата. Част от тях са само на английски, защото не ги знам на български:

1. Корен (root node) – това е най-горният връх, който считаме, че няма родител (или алтернативно е родител на себе си). Коренът в пример 1 е „Ръководител проект“.
2. Дете (child node) на даден връх – връх, който е свързан с дадения и е точно под него. „Тестер 1“ и „Тестер 2“ са деца на „Ръководител тестери“.
3. Родител (parent node) на даден връх – върхът, който е свързан с дадения и е точно над него. Всеки връх има точно един родител! „Ръководител проект“ е родител на „Дизайнер“.
4. Листо (leaf node) – всеки връх, който няма деца. „Програмист 1, 2, 3“, „Дизайнер“ и „Тестер 1, 2“ са листата в горното дърво.
5. Вътрешен връх (internal node) – всеки връх, който не е листо (те. има поне едно дете).
6. Level of a node – броя на ребрата в пътя от корена до дадения връх (в кое да е дърво, между всеки два върха има точно един прост път). „Ръководител проект“ е на ниво 0; „Ръководител програмисти“, „Дизайнер“ и „Ръководител тестери“ са на ниво 1; останалите са на ниво 3.

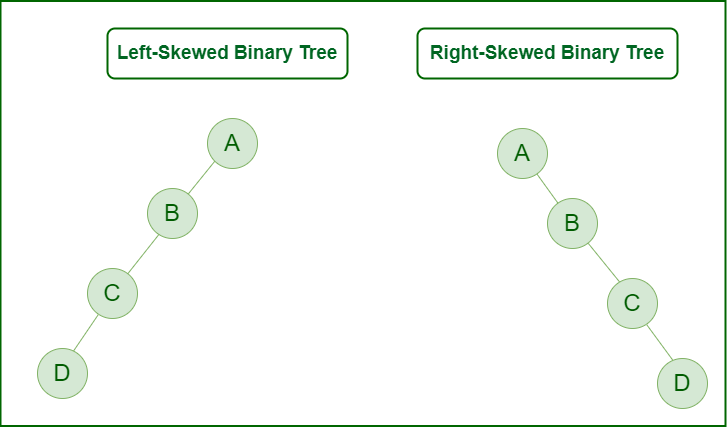
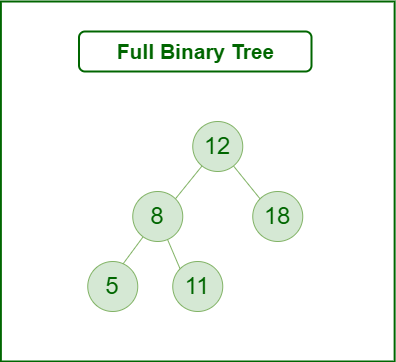
# Видове дървета по отношение на броя наследници

1. Двоично дърво: всеки връх на дървото има най-много два наследника (това са най-често използваните дървета).
2. N-тично дърво: всеки връх има най-много N на брой наследници (N може да е 2, 3, 4 … 100 и т.н.

# Видове двоични дървета

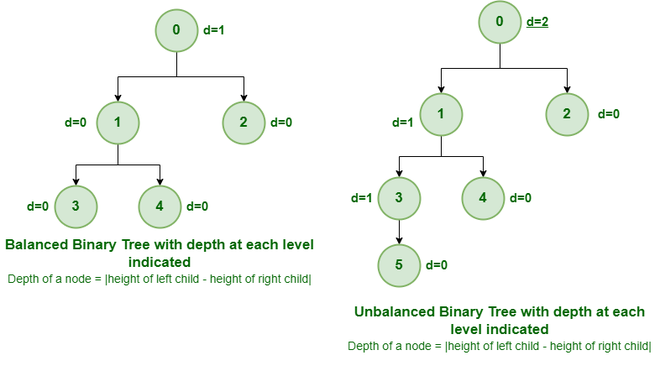
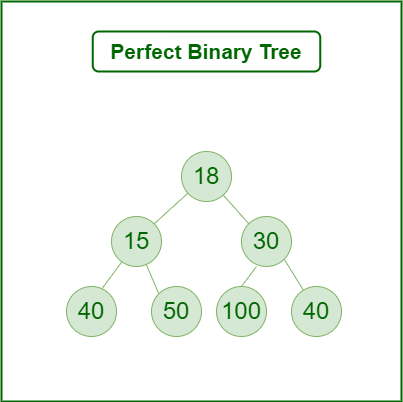
## По отношение на броя на наследниците

1. Full Binary Tree – всеки връх има или 0 или 2 деца.
2. Degenerate Binary Tree (изродено дърво) – дърво, чиито вътрешни върхове имат само по едно дете. Тогава дървото на практика става свързан списък и времевата сложност на основните операции е като на свързан списък.
3. Skewed Binary Trees – изродено дърво, чието ляво или дясно поддърво е доминиращо.



## По отношение на завършеност на нивата

1. Complete Binary Tree – всички нива са пълни с изключение може би на последното и върховете на последното ниво са възможно най-вляво.
2. Perfect Binary Tree – всички вътрешни върхове имат 2 деца и всички листа са на едно ниво. Броят на върховете на Perfect Binary Tree с N на брой нива е точно 2N – 1.
3. Balanced Binary Tree – двоично дърво, което е **балансирано по някакъв критерии** (на пример weight balanced, height balanced).



## Други видове

1. Binary Search Tree (двоично дърво за търсене) – лявото поддърво на всеки връх съдържа елементи строго по-малки от този във върха. Дясното поддърво на всеки връх съдържа елементи строго по-големи от този във върха.
   1. Отбелязвам, че при този вид дърво е нужно елементите да са сравними.
   2. Всички елементи трябва да са различни. Това ограничение може да се преодолее, ако на пример във всеки връх сложим брояч, за броя на елементите с тази стойност.
   3. Ако дървото не е изродено (те. височината му е от порядъка LogN), времевата сложност на намиране, добавяне и премахване на елемент е LogN.
   4. Намирането на елемент става напълно очевидно като тръгваме по лявото или дясното поддърво, в зависимост от това дали търсеният елемент е по-малък или по-голям от елемента в текущия връх.
2. AVL Tree – при добавяне и премахване на елементи от дърво може то да се изроди до свързан списък, което драстично влошава времевата сложност на операциите с дървото. AVL Tree (кръстено на създателите си: Adelson-Velsky и Landis) е самобалансиращо се двоично дърво за търсене балансирано по височина (height balanced). Има алгоритми, които след всяко добавяне/изтриване на връх проверяват дали дървото е балансирано и го балансират при нужда.
3. Red Black Tree (червено-черно дърво) – друг вид самобалансиращо се двоично дърво за търсене.

