

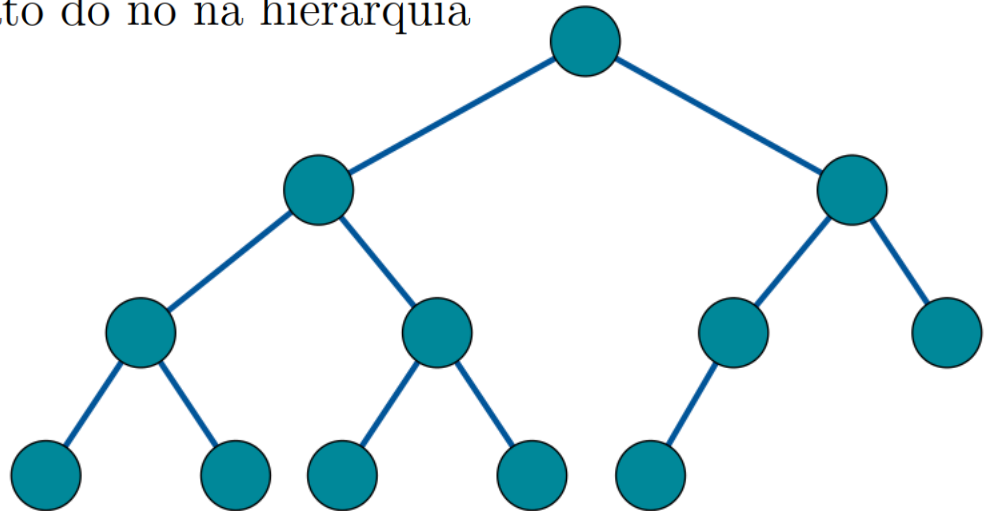
ÁRVORES BINÁRIAS (QUASE) COMPLETAS

Algoritmos e Estruturas de
Dados II

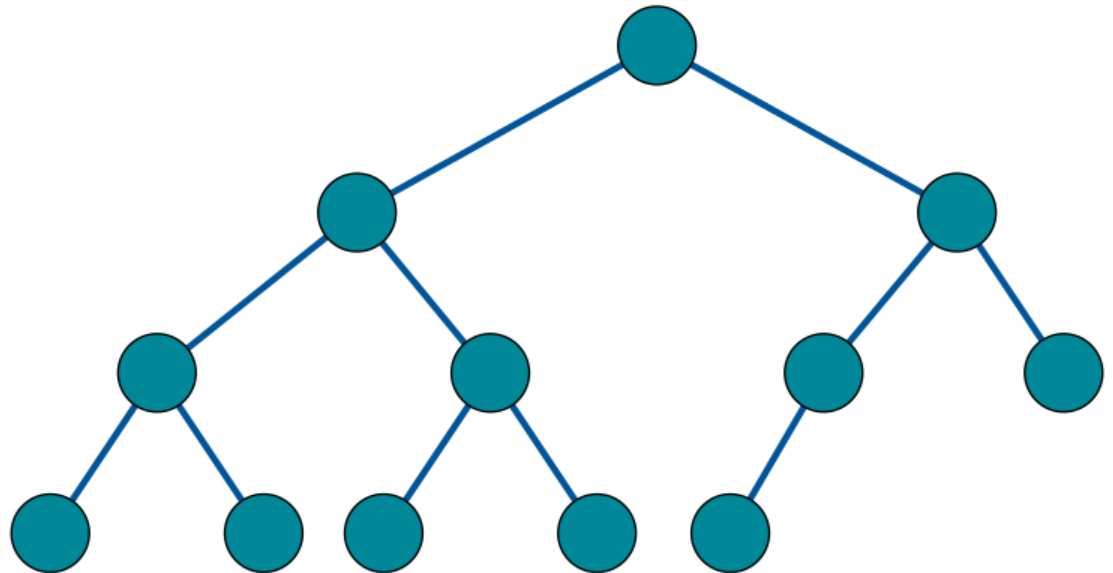
Prof. André Vignatti

Árvore: abstração de uma estrutura hierárquica (tipo abstrato de dados)

- nodos (nós, vértices): elementos da árvore (podem ser rotulados ou não)
- árvores podem ser enraizadas ou não
- raiz: início de hierarquia
- pai de um nó: o ancestral imediato na hierarquia
- filho de um nó: descendente imediato do nó na hierarquia
- folha: nó sem filho

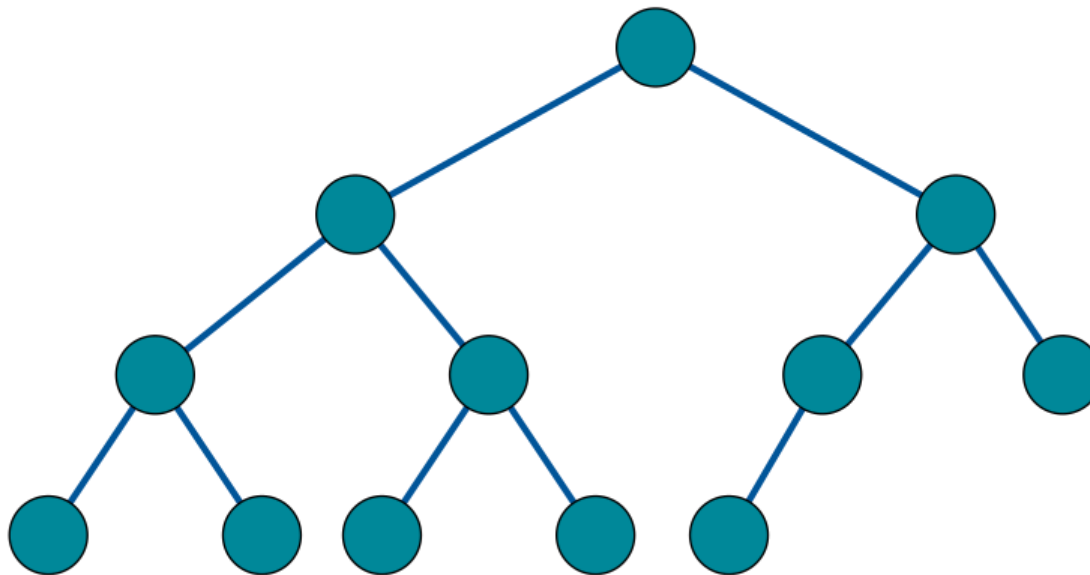


- nível k : nós com k ancestrais até a raiz.
- altura da árvore: quantidade de níveis
- altura de um nó: distância do nó até a folha mais longe
- árvore binária: todo nó tem no máximo dois filhos
- árvore binária completa: todo nível é completamente preenchido



Uma árvore binária de profundidade h é **quase completa** se:

1. É uma árvore binária completa até o nível $h - 1$.
2. No nível h , (o último nível), se um nó está presente, então todos os nós à esquerda desse nó também devem estar presentes.



An aerial photograph of a suburban neighborhood. The houses are arranged in a grid-like pattern with winding streets. The roofs are mostly light-colored, and the yards are green with many trees. The overall scene is a dense, well-maintained residential area.

PROPRIEDADES DE ÁRVORES BINÁRIAS (QUASE) COMPLETAS

Teorema. *O número de nós no nível k de uma árvore binária completa é 2^k*

Teorema. *O número de nós no nível k de uma árvore binária completa é 2^k*

Demonstração.

O nível 0 (raiz) tem 1 nó.

Todo nó não folha tem dois filhos.

Então,

a cada nível, o número de nós dobra em relação ao nível anterior.



Seja n o número de nós de uma árvore e h a sua altura

Teorema. *Em uma árvore binária completa, $n = 2^{h+1} - 1$*

Seja n o número de nós de uma árvore e h a sua altura

Teorema. *Em uma árvore binária completa, $n = 2^{h+1} - 1$*

Demonstração.

A quantidade de nós é igual a soma de nós em cada nível, i.e.

$$2^0 + 2^1 + \dots + 2^h$$

Usando a fórmula da p.g. finita, obtemos o resultado desejado.



Teorema. *Em uma árvore binária quase completa, $n \geq 2^h$ e $n \leq 2^{h+1} - 1$*

Teorema. *Em uma árvore binária quase completa, $n \geq 2^h$ e $n \leq 2^{h+1} - 1$*

Demonstração. Uma árvore binária quase completa é uma árvore binária completa sem alguns dos nós do último nível (nível h). Assim, partindo de uma árvore binária completa, podemos retirar no máximo $2^h - 1$ nós sem alterar a altura da árvore, obtendo assim uma árvore binária quase completa.

Desta forma, o número de nós da árvore binária quase completa é pelo menos

$$2^{h+1} - 1 - (2^h - 1) = 2^{h+1} - 2^h = 2^h.$$



Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Demonstração.

Se a árvore é completa, então $n = 2^{h+1} - 1$

Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Demonstração.

Se a árvore é completa, então $n = 2^{h+1} - 1$

Isolando h , temos

$$h = \log_2(n + 1) - 1$$

Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Demonstração.

Se a árvore é completa, então $n = 2^{h+1} - 1$

Isolando h , temos

$$h = \log_2(n + 1) - 1$$

Note que $\log_2(n + 1)$ é inteiro (pois, pelo Teorema anterior, $n = 2^{h+1} - 1$)

Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Demonstração.

Se a árvore é completa, então $n = 2^{h+1} - 1$

Isolando h , temos

$$h = \log_2(n + 1) - 1$$

Note que $\log_2(n + 1)$ é inteiro (pois, pelo Teorema anterior, $n = 2^{h+1} - 1$)

podemos escrever $\log_2(n + 1) = \lfloor \log_2 n \rfloor + 1$

Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Demonstração.

Se a árvore é completa, então $n = 2^{h+1} - 1$

Isolando h , temos

$$h = \log_2(n + 1) - 1$$

Note que $\log_2(n + 1)$ é inteiro (pois, pelo Teorema anterior, $n = 2^{h+1} - 1$)

podemos escrever $\log_2(n + 1) = \lfloor \log_2 n \rfloor + 1$

Portanto,

$$h = (\lfloor \log_2 n \rfloor + 1) - 1 = \lfloor \log_2 n \rfloor$$

Teorema. *Em uma árvore binária (quase) completa, $h = \lfloor \log_2 n \rfloor$*

Demonstração.

Se a árvore é completa, então $n = 2^{h+1} - 1$

Isolando h , temos

$$h = \log_2(n + 1) - 1$$

Note que $\log_2(n + 1)$ é inteiro (pois, pelo Teorema anterior, $n = 2^{h+1} - 1$)

podemos escrever $\log_2(n + 1) = \lfloor \log_2 n \rfloor + 1$

Portanto,

$$h = (\lfloor \log_2 n \rfloor + 1) - 1 = \lfloor \log_2 n \rfloor$$

Finalmente, a altura da árvore quase completa é igual a altura da árvore completa.





IMPLEMENTAÇÃO DE ÁRVORES BINÁRIAS QUASE-COMPLETAS

A implementação deste caso particular de árvores usa vetor.

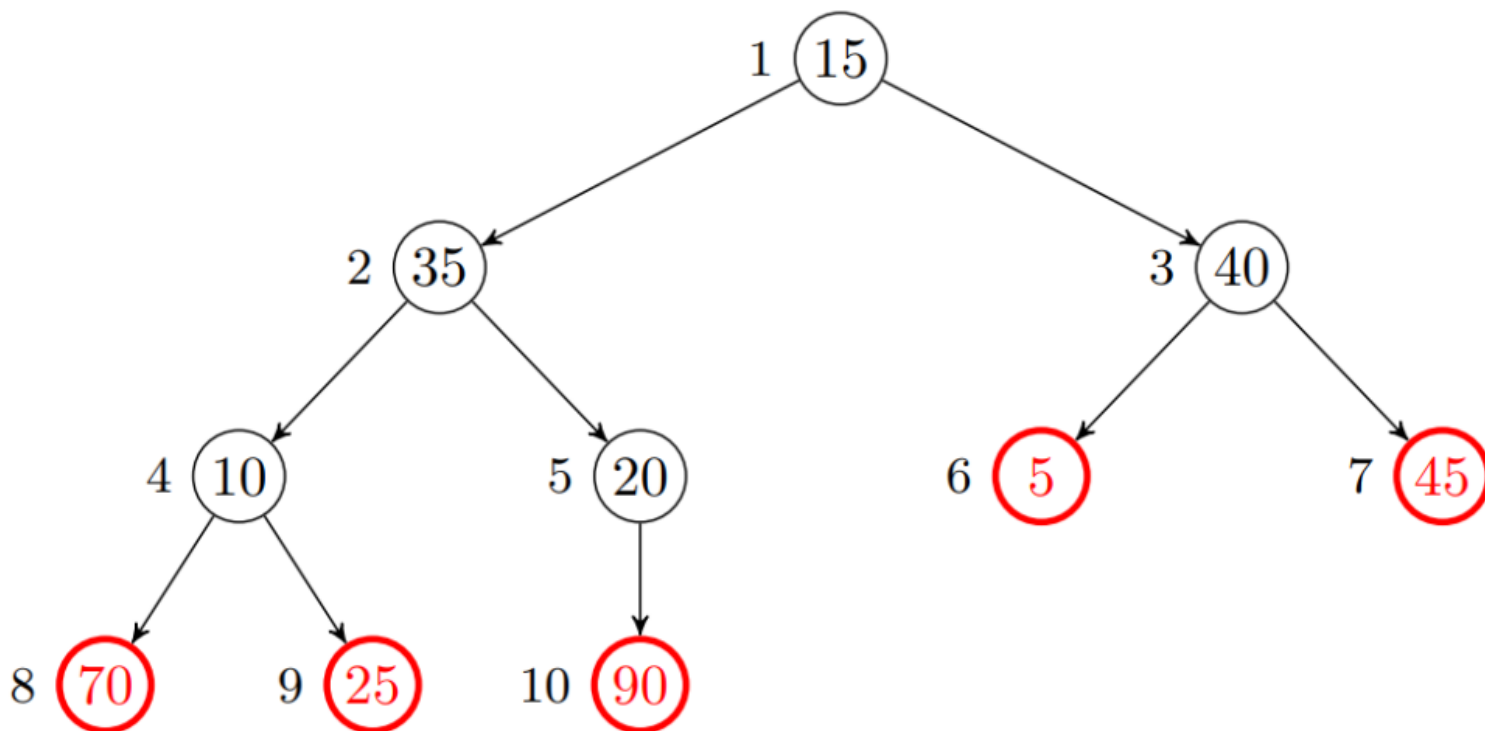
O primeiro elemento do vetor (índice 1) contém a raiz.

Todos os outros elementos têm a seguinte propriedade:

- O pai do nodo i está no índice $i/2$ (divisão inteira)
- O filho da esquerda do nodo i está no índice $2i$
- O filho da direita do nodo i está no índice $2i + 1$

Estas operações são extremamente rápidas nos computadores.

	1	2	3	4	5	6	7	8	9	10
v	15	35	40	10	20	5	45	70	25	90



$$\text{Teorema.} \quad \sum_{i=0}^k 2^i \cdot i = k2^{k+2} - (k+1)2^{k+1} + 2$$

Demonstração.

Lembrando, a fórmula da soma de p.g. finita é

$$\sum_{i=0}^k c^i = \frac{c^{k+1} - 1}{c - 1}$$

$$\frac{d(u/v)}{dx} = \frac{v\left(\frac{du}{dx}\right) - u\left(\frac{dv}{dx}\right)}{v^2}$$

Derivando os dois lados e multiplicando por c

$$\sum_{i=0}^k c^i i = \frac{kc^{k+2} - (k+1)c^{k+1} + c}{(c-1)^2}$$

Fazendo $c = 2$, obtemos o resultado desejado.

