

Trees

Svarny Petr

Katedra logiky FF UK

29. března 2021

Overview

Stromy

Implementace stromu

Pruchody stromem

Vyhledavani, BST a operace na nem

Stromy

Volný Strom

Souvislý acyklický neorientovaný graf.

Les

Acyklický neorientovaný graf.

Vlastnosti stromu

Věta

Nechť $G = (V(G), E(G))$ je neorientovaný graf. Pak jsou následující tvrzení ekvivalentní:

- ▶ G je volný strom.
- ▶ Jakékoliv 2 uzly v G jsou spojeny jedinečnou cestou.
- ▶ G je souvislý, ale pokud vyjmeme jakoukoliv hranu z $E(G)$, graf bude nesouvislý.
- ▶ G je souvislý a $|E(G)| = |V(G)| - 1$.
- ▶ G je acyklický a $|E(G)| = |V(G)| - 1$.
- ▶ G je acyklický, ale pokud přidáme jednu hranu, bude graf obsahovat alespoň jeden cyklus.

Korenove stromy

Definice: Kořenový strom

- ▶ **Kořenový strom** je volný strom, kdy jeden z uzlů je označen jako **kořen/root** r .
- ▶ Uzly u ve vzdálenosti d od kořenu r tvoří hladinu uzlů v hloubce/depth či height $h(u) = d$, kořen má hloubku $h(r) = 0$.
- ▶ Uzly u, v za sebou na cestě z r , potom u je **předek**/ancestor a v je **potomek**/successor.
- ▶ Nejbližší předek = **rodič**/parent, nejbližší potomek je **syn** nebo **děti**/child.
- ▶ **Sourozenci**/siblings = uzly se stejným rodičem.
- ▶ **Stupeň**/degree počet dětí uzlu.
- ▶ **List** nebo **vnější uzel**/leaf nebo terminální uzel je uzel stupně 0.
- ▶ k -ární strom = každý uzel má stupeň **nejvýše** k .

Binární kořenový strom

Definice: Binární kořenový strom

- ▶ **Binární kořenový strom**, tj. kořenový strom s uzly co mají stupeň max 2.
- ▶ **Podstrom**/subtree, tj. strom pod daným uzlem.

Pravidelný a vyvážený binární strom

Definice: Pravidelný binární strom

Pravidelný binární strom/full binary tree, tj. kořenový strom s uzly co mají stupeň 0 nebo 2.

Definice: Vyvážený binární strom

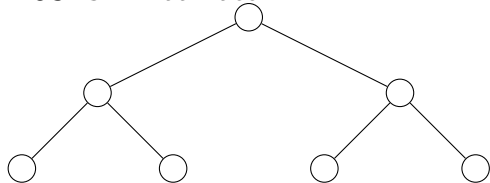
Vyvážený binární strom/complete, tj. hloubka všech listů je stejná.

Důsledek

Pro $n =$ počet uzlů, hloubka $h(l) = \log_2(n)$ neboli $(2^{h+1} - 1) = n$.

Binarni strom online

=USFCA vizualizace=



Implementace binárního stromu

Nějaké nápady, jak implementovat binární strom?

Implementace binárního stromu

Key: levý a pravý potomek.

```
class Node:
```

```
    def __init__(self, data):
```

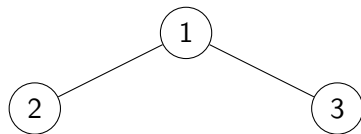
```
        self.left = None
```

```
        self.right = None
```

```
        self.data = data
```

Implementace binarniho stromu

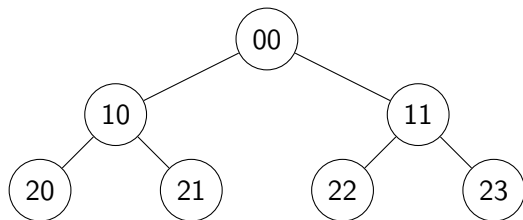
```
> root = Node(1)  
> root.left = Node(2)  
> root.right = Node(3)
```



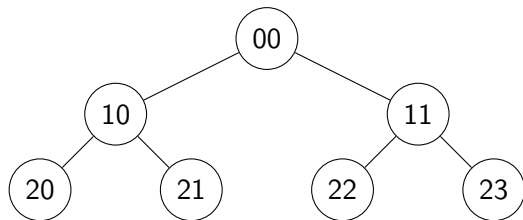
Uziti stromu

- ▶ HTML/XML objektový model,
- ▶ Parsovací stromy při kompilaci,
- ▶ Heap (Halda) a její aplikace ...

Ukazkový strom

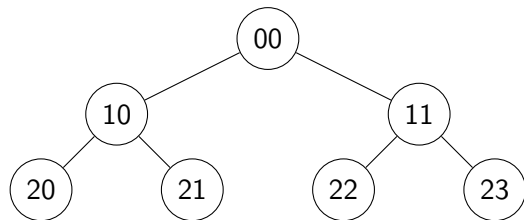


InOrder



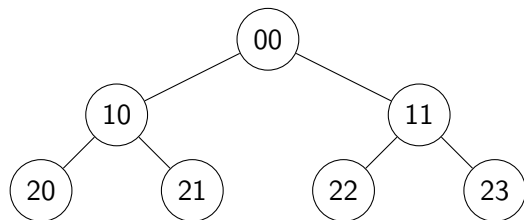
20 10 21 00 ...

InOrder



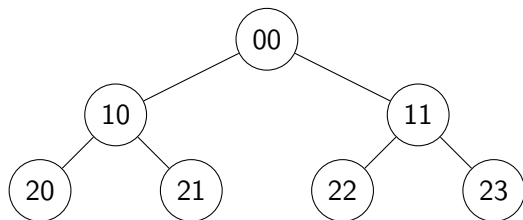
20 10 21 00 22 11 23

InOrder



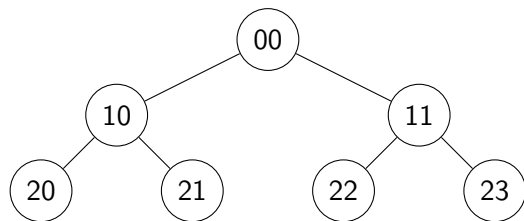
20 10 21 00 22 11 23 = "v pořadí projekce"

PreOrder



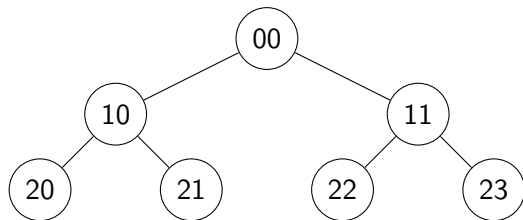
Od kořene podél levé hrany...

PreOrder



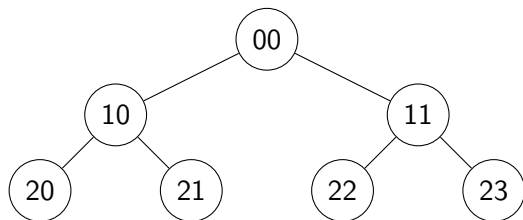
00 10 20 21 ...

PreOrder



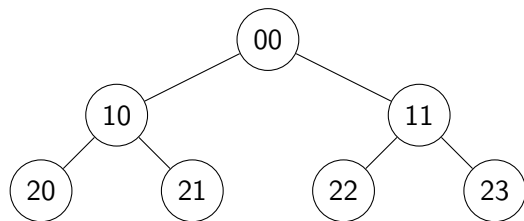
00 10 20 21 11 22 23

PostOrder



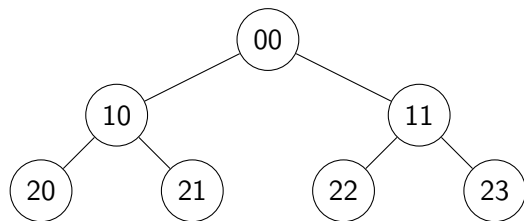
Od levé strany od listů...

PostOrder



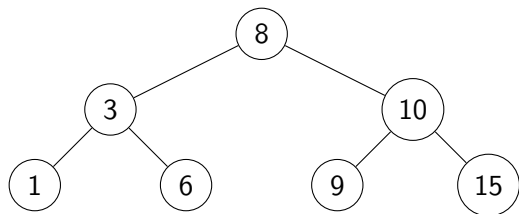
20 21 10 22 ...

PostOrder



20 21 10 22 23 11 00

Binarni vyhledavaci strom



Binární vyhledávací strom

Definice: Binární vyhledávací strom

?

Binární vyhledávací strom

Definice: Binární vyhledávací strom

Binární vyhledávací strom/Binary search tree (BST) je binární strom s jedinečnými uzly, kdy levý podstrom daného uzlu u obsahuje uzly s hodnotami menšími než je daný uzel u a pravý podstrom obsahuje uzly s hodnotami většími než je daný uzel u . A každý podstrom je opět BST.