

# DELTA TopGun

## 07 - Stromové struktury

Tomáš Faltejsek, Luboš Zápotočný, Michal Havelka

2022

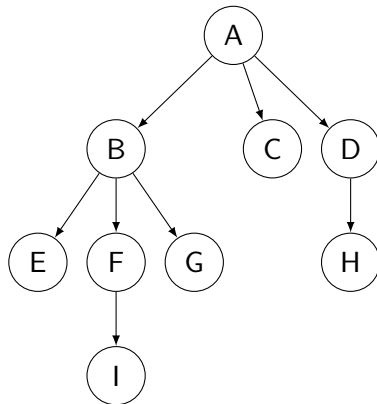
# Hlavní motivace

- Vyhledávání sekvenční  $\times$  lineární datové struktury
- Přirozená reprezentace
- Rozšíření konceptu linked list

# Stromová struktura



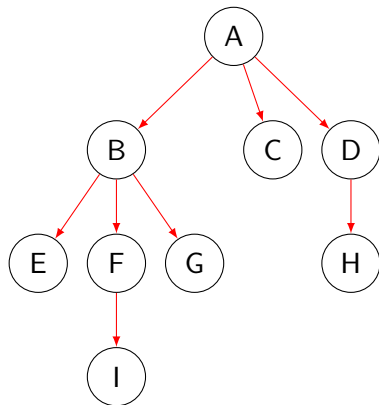
# Stromová struktura



# Vlastnosti stromové struktury

- **Acyklický** orientovaný graf
- Skládá se z **uzlů** které jsou propojeny orientovanými **hranami**
- Rekurzivní datová struktura (podstromy)

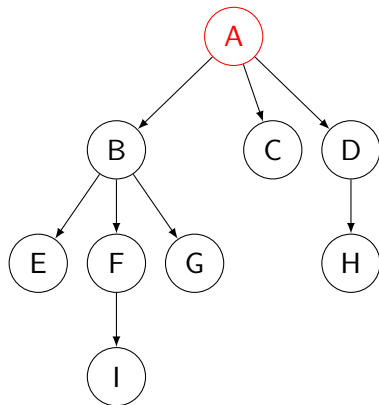
# Vlastnosti stromové struktury – hrany (edge)



## Vlastnosti

- Každý uzel (s **výjimkou kořene**) je propojen orientovanou hranou přesně z jednoho dalšího uzle
- Orientace hrany je rodič (*parent*) → dítě (*child*)

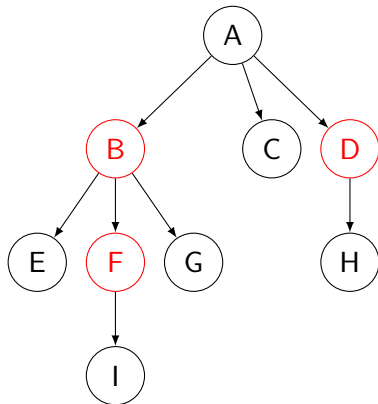
# Vlastnosti stromové struktury – kořen (**root**)



## Vlastnosti

- Kořen **nemá rodiče**
- **A** je rodičem B, C, D
- Z kořene se dostaneme k libovolnému uzlu

# Vlastnosti stromové struktury – uzel (**inner node**)



## Vlastnosti

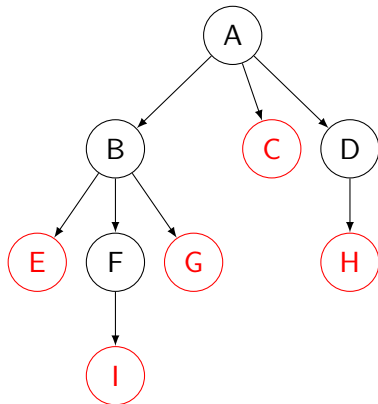
- Každý uzel může mít *libovolný* počet potomků
- **Inner node** musí mít alespoň jednoho potomka
- **Inner node** má právě jednoho rodiče



# Vlastnosti stromové struktury – uzel (inner node)

```
// Node of a general tree structure with n children
typedef struct Node {
    int data; // Or any other data-type
    Node * child1;
    Node * child2;
    // .
    // .
    // .
    Node * childN;
} Node;
```

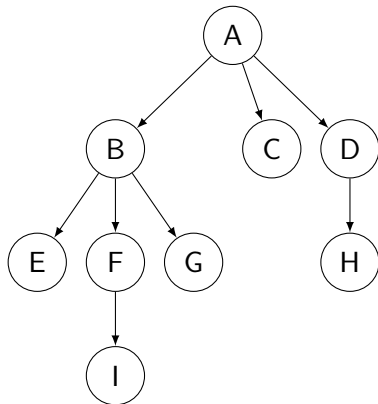
# Vlastnosti stromové struktury – list (leaf node)



## Vlastnosti

- *Leaf node* má právě jednoho rodiče
- *Leaf node* nemá **žádného** potomka

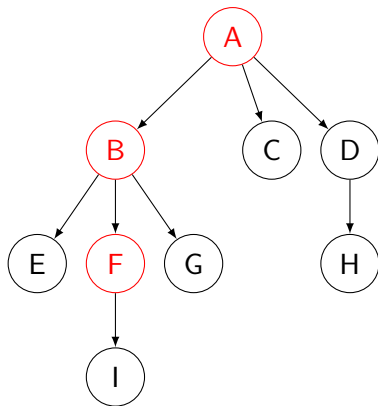
# Vlastnosti stromové struktury – předek (**ancestor**)



## Vlastnosti

- Libovolný přecházející *uzel* v uspořádané stromové struktuře

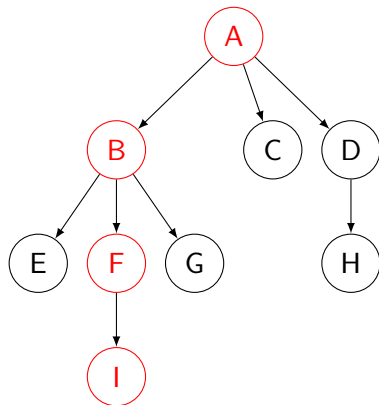
# Vlastnosti stromové struktury – hloubka uzlu (depth of node)



## Vlastnosti

- Počet hran od *kořene* k *uzlu*
- **Hloubka** uzlu **F** je 2

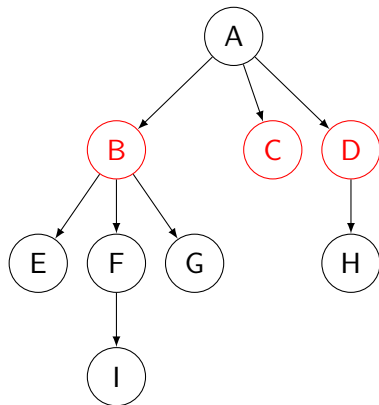
# Vlastnosti stromové struktury – výška (**height**)



## Vlastnosti

- Počet hran od *uzlu* k nejhlubšímu *listu*
- Hloubka stromu je **3**

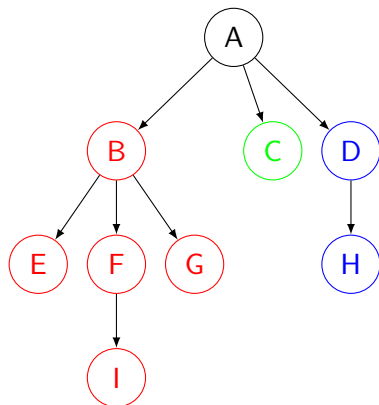
# Vlastnosti stromové struktury – sourozenci (**siblings**)



## Vlastnosti

- Uzly se stejným rodičem (*parent*)

# Vlastnosti stromové struktury – podstrom (subtree)



## Vlastnosti

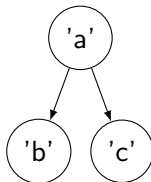
- Kořen **A** má **3 podstromy**

# Vlastnosti stromové struktury – shrnutí

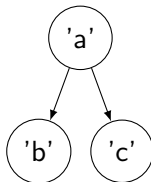
- 1 Acyklický graf
- 2 Orientované hrany
- 3 Přesně jeden *kořen*
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče
- 5 Každá *inner* node má alespoň jednoho potomka
- 6 *Leaf node* nemá žádného potomka



## Jedná se o stromovou strukturu? – Příklad 1



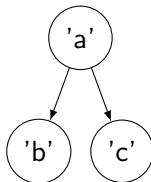
# Jedná se o stromovou strukturu? – Příklad 1



## Řešení

- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden *kořen* ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✓
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf node* nemá žádného potomka ✓

# Jedná se o stromovou strukturu? – Příklad 1

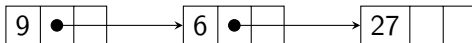


## Řešení

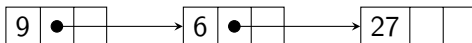
- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden *kořen* ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✓
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

→ **Validní stromová struktura**

## Jedná se o stromovou strukturu? – Příklad 2



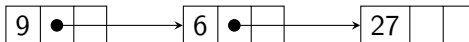
# Jedná se o stromovou strukturu? – Příklad 2



## Řešení

- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden *kořen* ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✓
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

# Jedná se o stromovou strukturu? – Příklad 2

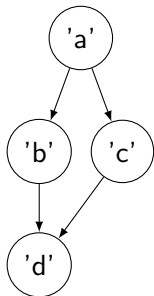


## Řešení

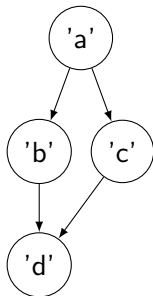
- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden *kořen* ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✓
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

→ **Validní stromová struktura** s branching factorem = 1 (unární strom)

# Jedná se o stromovou strukturu? – Příklad 3



# Jedná se o stromovou strukturu? – Příklad 3

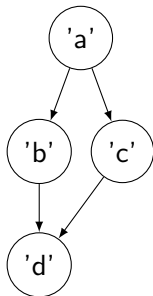


## Řešení

- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden kořen ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✗
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓



# Jedná se o stromovou strukturu? – Příklad 3

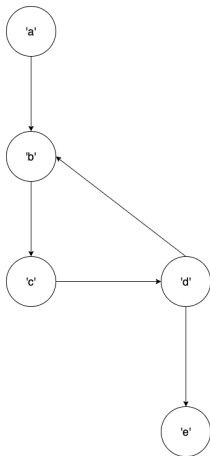


## Řešení

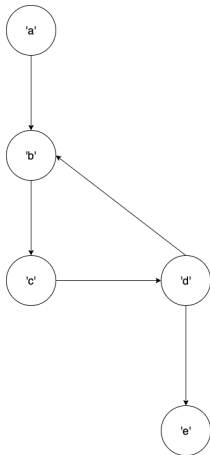
- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden kořen ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✗
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

→ **NEVALIDNÍ stromová struktura**

# Jedná se o stromovou strukturu? – Příklad 4



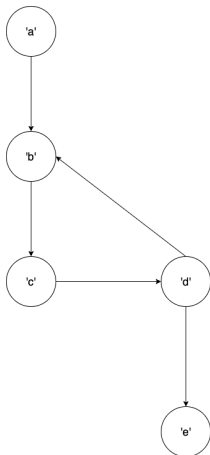
# Jedná se o stromovou strukturu? – Příklad 4



## Řešení

- 1 Acyklický graf ✗
- 2 Orientované hrany ✓
- 3 Přesně jeden kořen ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✗
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

# Jedná se o stromovou strukturu? – Příklad 4



## Řešení

- 1 Acyklický graf ✗
- 2 Orientované hrany ✓
- 3 Přesně jeden kořen ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✗
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

→ **NEVALIDNÍ stromová struktura**

## Jedná se o stromovou strukturu? – Příklad 5

9

# Jedná se o stromovou strukturu? – Příklad 5

9

## Řešení

- 1 Acyklický graf ✓
- 2 Orientované hrany ✓
- 3 Přesně jeden *kořen* ✓
- 4 Každý *inner* a *leaf* node má přesně jednoho rodiče ✓
- 5 Každá *inner* node má alespoň jednoho potomka ✓
- 6 *Leaf* node nemá žádného potomka ✓

# Jedná se o stromovou strukturu? – Příklad 5

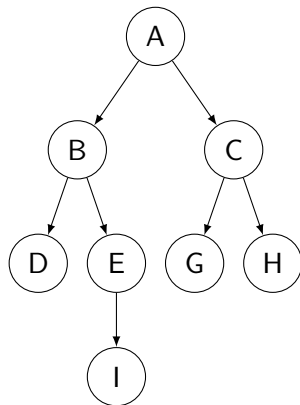
9

## Řešení

- ① Acyklický graf ✓
- ② Orientované hrany ✓
- ③ Přesně jeden *kořen* ✓
- ④ Každý *inner* a *leaf* node má přesně jednoho rodiče ✓
- ⑤ Každá *inner* node má alespoň jednoho potomka ✓
- ⑥ *Leaf* node nemá žádného potomka ✓

→ **Validní stromová struktura**

# Binární strom



## Vlastnosti

- Každý uzel má **maximálně dva** potomky



# Binární strom - node

```
// Node of a binary tree
typedef struct Node {
    int data; // Or any other data-type
    Node * left;
    Node * right;
} Node;
```

# Průchody - INORDER - LNR

- 1 Rekurzivní průchod levým podstromem aktuálního uzlu
- 2 Navštívení aktuálního uzlu
- 3 Rekurzivní průchod pravým podstromem aktuálního uzlu

# Průchody - POSTORDER - LRN

- Rekurzivní průchod levým podstromem aktuálního uzlu
- Rekurzivní průchod pravým podstromem aktuálního uzlu
- Navštívení aktuálního uzlu

# Průchody - PREORDER - NLR

- Navštívení aktuálního uzlu
- Rekurzivní průchod levým podstromem aktuálního uzlu
- Rekurzivní průchod pravým podstromem aktuálního uzlu

# Ukázka - syntaktické stromy

– **Demonstrace na tabuli** –

# Přirozeně hierarchická data - file system

– **Demonstrace na tabuli** –

# Organizace dat - Binární vyhledávací strom

**Na příští přednášce**