ADT Strom

- Strom je datová struktura, která uchovává objekty (prvky) hierarchicky ve vztahu předchůdce-následovník (otec-syn)
- Stromy patří mezi často používané datové struktury v mnoha oblastech computer science
- Příklady použití :
  - reprezentace znalostí, stavového prostoru v umělé inteligenci
  - popis scény v oblasti zpracování a analýza obrazu, počítačová grafika
  - vyhledávací stromy v databázových systémech
  - rozhodovací stromy – expertní systémy
  - organizace adresářů a souborů v souborovém systému OS,
  - komprese dat (Hufmannovy kódovací stromy, fraktálová komprese)
  - atd.

Definice stromu

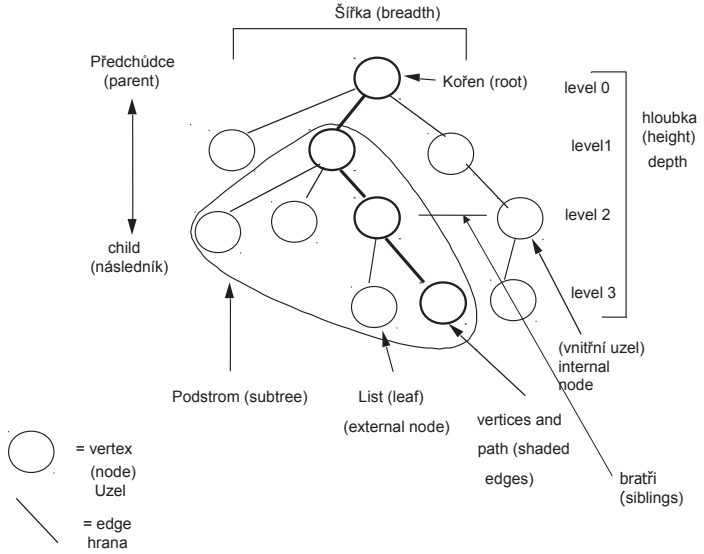
Def. 1. : Strom T je konečná množina jednoho nebo více prvků (uzlů), z nichž jeden je označen jako ***r*** kořen (root) a zbývající uzly jsou rozděleny do $n \geq 0$ disjunktních podmnožin $T_1, T_2, \ldots T_k$, které jsou také stromy a jejichž kořeny $r_1, r_2, \ldots, r_k$ jsou následníky kořene ***r*** .

Def. 2 : Strom je graf, ve kterém existuje pouze jedna cesta z uzlu ***r*** (kořene), do kteréhokoliv dalšího uzlu grafu (neobsahuje cykly)

Def. 3:

- Strom je prázdný,
- (nebo) strom obsahuje jediný uzel (kořen)
- (nebo) strom obsahuje kořen spojený s jedním nebo více podstromy

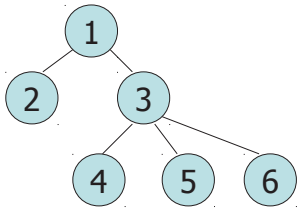
Základní terminologie

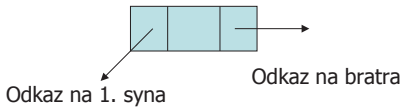


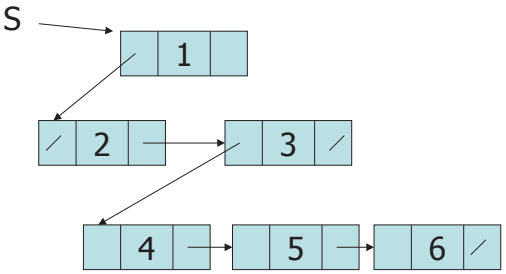
- **Kořen** - uzel který nemá předchůdce, ve stromu může být pouze jediný kořen
- **Listy** (vnější uzly) – uzly které nemají žádného následníka
- **Vnitřní uzly** – uzly které mají alespoň jednoho následníka
- **Cesta** – je-li $n_1, n_2, \ldots n_k$ množina uzlů ve stromu takových, že n_i je předchůdce n_{i+1}, pro $1 \leq i \leq k$, pak se tato množina nazývá cesta z uzlu n_1 do n_k
- **Délka cesty** – počet hran, které spojují uzly cesty
- **Hloubka uzlu** – délka cesty od kořene do uzlu
- **Výška stromu** – délka cesty od kořene k nejhlubšímu uzlu (největší hloubka)

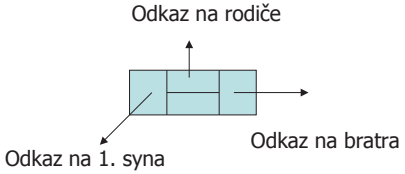
Typy stromů a implementace

- Obecný strom - vnitřní uzly stromu mohou mít libovolný počet následníků. Implementuje se výhradně jako dynamická struktura – seznam zřetězených prvků.

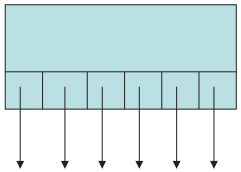






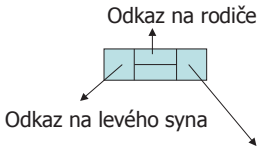


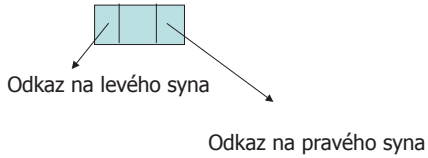
- n-ární strom – vnitřní uzly stromu obsahují maximálně n následníků. Implementuje se obvykle jako seznam zřetězených prvků, kde každý uzel stromu obsahuje pole ukazatelů na následníky (syny)



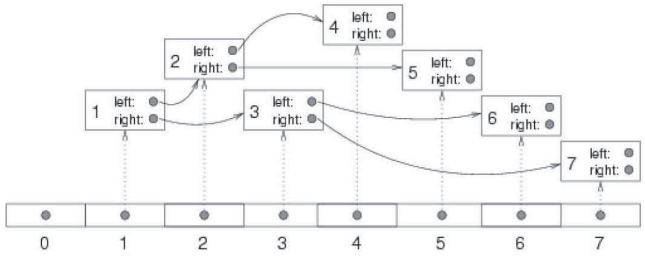
Odkazy na syny

- binární strom – speciální případ n-árního stromu, každý vnitřní uzel může mít nejvýše dva následníky (syny). Implementuje se jako seznam zřetězených prvků, někdy jako pole.





- Implementace binárního stromu jako vektor

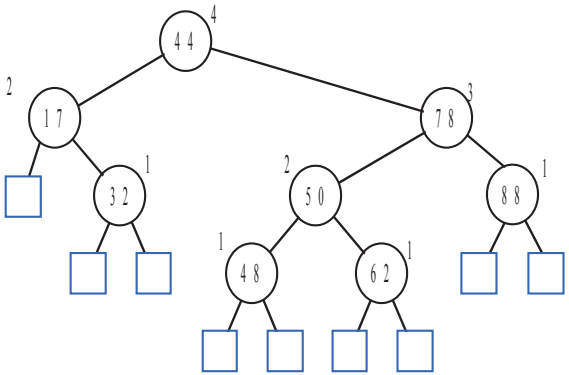


AVL strom

Adelson-Velskii, Landis

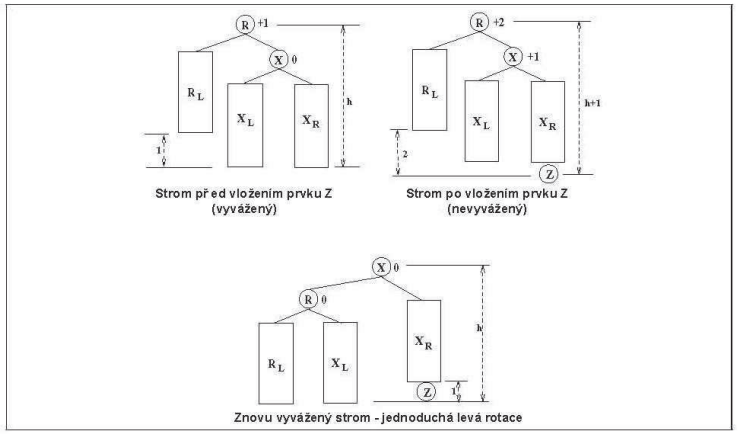
(vyvážený strom)

AVL strom je výškově vyvážený binární vyhledávací strom, pro který platí, že pro libovolný vnitřní uzel stromu se výška levého a pravého syna liší nejvýše o 1

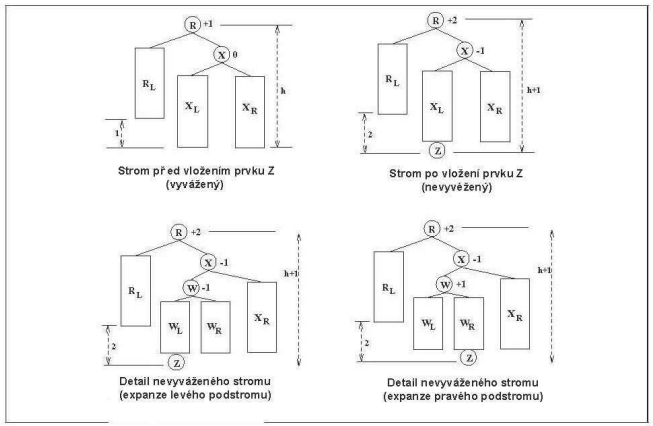


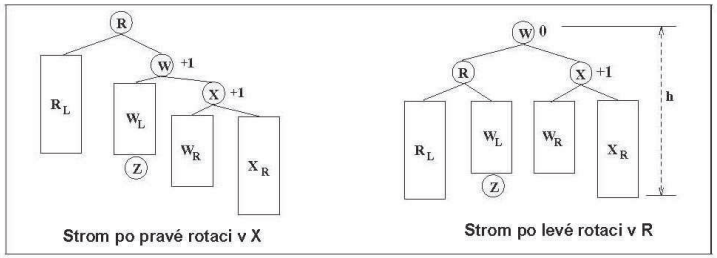
- Vyváženost AVL stromu se kontroluje po každé operaci vložení a zrušení prvku, v případě že je vyváženost porušena, provádí se opětovné vyvážení pomocí jedné popř. několika rotací v jednotlivých částech stromu.
- Implementace je obdobná jako u BVS, datová struktura pro uzel stromu je doplněna o celočíselnou proměnnou reprezentující stupeň vyváženosti uzlu, který může nabývat následujících hodnot:
  - 0 – oba podstromy jsou stejně vysoké
  - 1 – pravý podstrom je o 1 vyšší
  - 2 – pravý podstrom je o 2 vyšší
  - -1 – levý podstrom je o 1 vyšší
  - -2 – levý podstrom je o 2 vyšší
- Rotace :
  - Jednoduchá pravá (levá) – používáme pokud vyvažujeme přímou větev, tj. jsou-li znaménka stupně vyváženosti stejná
  - Dvojitá pravá (levá) – používá se tehdy pokud nejde použít jednoduchá rotace – vyvažujeme-li „zalomenou" větev.

- **Případ 1**: Pravý syn bude mít po operaci vložení hodnotu +1
  - Pokud měl uzel R před vložením prvku do pravého podstromu stupeň vyvážení +1, a pravý syn X uzlu R hodnotu 0 bude mít X po vložení prvku Z do pravého podstromu X_R hodnotu +1 a uzel R hodnotu +2. K opětovnému vyvážení použijeme levou rotaci v uzlu R



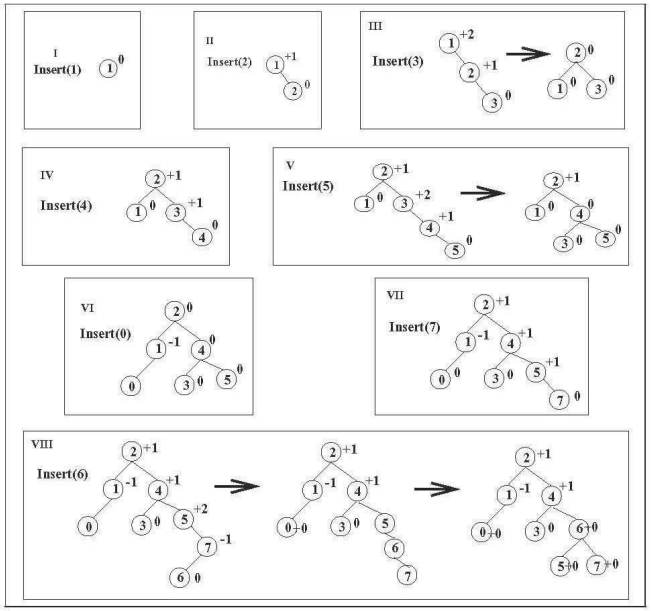
- Případ 2: Pravý syn bude mít po operaci vložení hodnotu -1
  - Stupeň vyvážení pravého podstromu uzlu R je v tomto případě -1, protože Z je vložen do X_L. Stupeň vyváženosti uzlu R se změní z +1 na +2. Ke opětovnému vyvážení je nutné provést 2 rotace, první je pravá rotace kolem X, druhá levá kolem R. Výška stromu zůstává po vyvážení stejná jako před vyvážením.





Algoritmus vložení prvku X do AVL stromu S

1. Vyhledávání X v S, dokud není zřejmé že prvek ve stromu S neexistuje
2. Zařazení X v místě, kde se ukončilo vyhledávání
3. Zpětný přepočet stupňů vyvážení S od přidaného uzlu vzhůru (od listu ke kořenu). V případě že došlo k rozvážení S (v místě, kde se poprvé vyskytne stupeň vyváženosti s hodnotou +2) se vykoná následující bod
4. Vyvážení S jednou ze dvou výše uvedených rotací



Vložení prvků {1,2,3,4,5,0,7,6} do AVL stromu

Zrušení prvků v AVL stromu

- Rušení prvků je podobné jako u BVS stromu, po zrušení prvku je nutné provést kontrolu stupně vyváženosti uzlů směrem ke kořeni a v případě že je to nutné znovu vyvážit strom pomocí jednoduchých popř. dvojnásobných rotací