

# Searching Trees

Svarny Petr

Katedra logiky FF UK

19. dubna 2021

# Overview

Blog o praxi

Binarni stromy

# Prakticky

=Practical data structures=

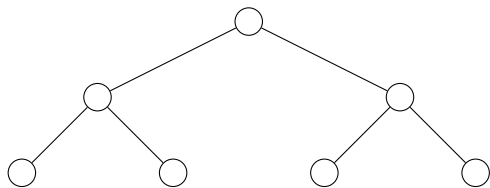
- ▶ Stromy (Skype, Uber...)
- ▶ Vážené grafy (Skyscanner)
- ▶ Hashtable (všude)
- ▶ atd.

# Binární vyhledávací strom

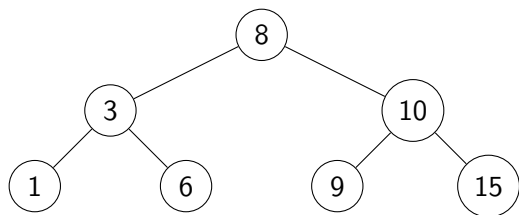
## Definice: Binární vyhledávací strom

**Binární vyhledávací strom**/Binary search tree (BVS) 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 BVS.

# Binarni vyhledavaci strom



# Binarni vyhledavaci strom

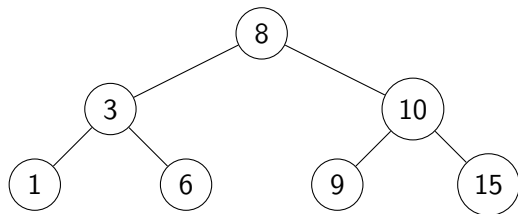


# Nalezení uzlu

Jak mohu vyhledávat uzel v Binárním vyhledávacím stromě?

# Nalezení uzlu

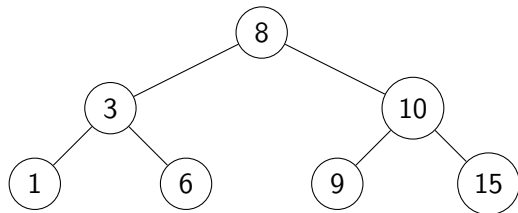
1. Začni v kořeni.
2. Vyhledávaný prvek nalezen nebo zvol podstrom: levý pro menší než kořen a pravý pro větší než kořen. Pokud již není podstrom, prvek není ve stromě.
3. Opakuj s podstromem.





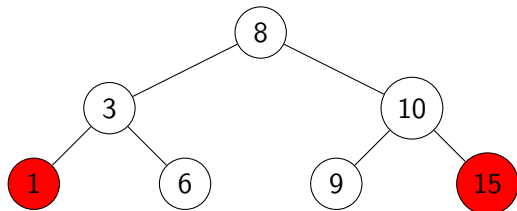
# Vlastnosti

- ▶ Kde je maximum v BVS?
- ▶ Kde je minimum v BVS?



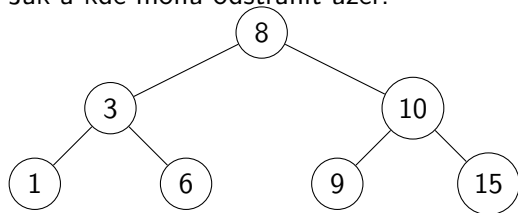
# Vlastnosti

- ▶ Maximum = nejvíc vpravo
- ▶ Minimum = nejvíc vlevo



# Odstraneni uzlu

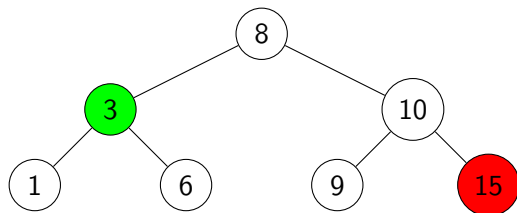
Jak a kde mohu odstranit uzlu?



# Odstranění uzlu

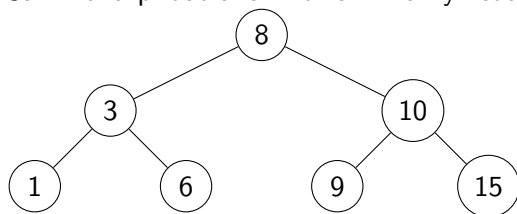
Jak a kde mohu odstranit uzel?

1. Uzel v listu odstraň.
2. Uzel s jedním potomkem, jen nahradit potomkem.
3. Uzel s dvěma potomky, nahrad' předchůdcem (následníkem) a vypusť předchůdce (následníka).



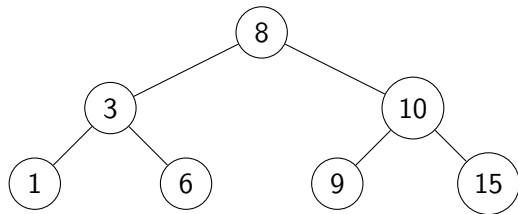
## Přidání uzlu

Jak mohu přidat uzel k binárnímu vyhledávacímu stromu? A kam?



# Pridani uzlu

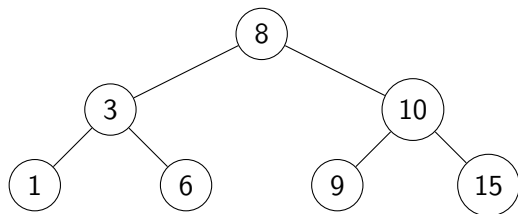
1. Iterativně
2. Rekurzivně



Viz =Techiedelight=

## Zakladni myslenska pridani uzlu

1. Najdi uzel, co chceš vložit.
2. V místě kde hledání skončilo, přidej nový uzel jako levý či pravý list podle velikosti. Pokud byl uzel nalezen, tak nelze přidat.



# Vyvažování stromu

Přidávání uzlů však narušuje vyváženost stromu a nevyvážený strom snižuje efektivitu prohledávání.

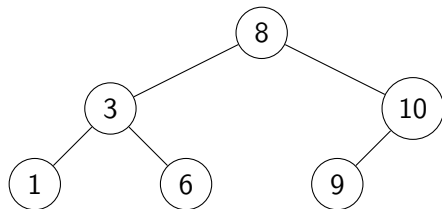
- ▶ Vyvážený strom  $O(\log_2 n)$
- ▶ Nevyvážený  $O(n)$
  
- ▶ Silná podmínka = shoda h, pro všechny je počet uzlů vlevo = počtu uzlů vpravo
- ▶ Slabá podmínka = je to skoro shodné
  - ▶ Adelson-Velsky a Landis strom (AVL)
  - ▶ Červeno-černý strom

Viz také =Wiki=



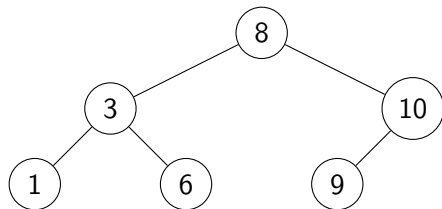
# AVL strom

- ▶ Pro každý uzel je rozdíl mezi hloubkou pravého a levého podstromu je max 1.



## AVL strom - pridani

- ▶ Přidat uzel jako obvykle.
- ▶ Vyvážit strom pomocí levé či pravé rotace od vloženého uzlu pokud je třeba.



=Geeks=

# Rotace - případy

Rozlišujeme případy podle připojení (Levý-Levý ap.)

LL Pravorotace prarodiče.

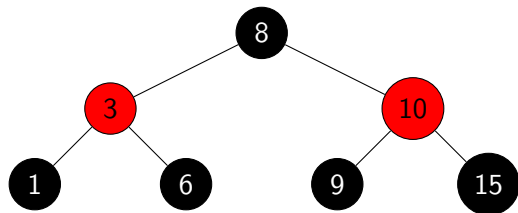
LR Levorotace rodiče a potom LL.

RR Levorotace prarodiče.

RL Pravorotace rodiče, potom RR.

# Červeno-černý strom

1. Každý uzel má jednu barvu (červenou nebo černou=).
2. Kořen je černý.
3. Červené uzly spolu nesousedí (tj. nemají potomka či rodiče červeného).
4. Každá cesta z uzlu k libovolnému jeho potomkovi-listu má stejný počet černých uzlů (černá výška).



=Cornell=

=Geeks=

# Červeno-černý strom - přidání

1. Přidáme uzel k listu jako obvykle. Obarvíme na červeno.
2. Upravíme dokud je porušeno pravidlo oddělení červených.
3. Ujistíme se, že kořen je černý.

Viz =USFCA vizualizace=

# Červeno-černý strom - případy přidání

1. Rodič je černý - vše ok.
2. Rodič je červený a
  - 2.1 "Strýc" je černý nebo není - rotace podle LL/LR/RR/RL scénáře.
  - 2.2 "Strýc" je červený - přebarvení rodiče a strýce a prarodiče (pokud je to kořen, tak také, přebarví se v 3. kroku, viz minulá stránka).

Viz =WISC=

# Cerveno-cerny strom vs AVL

Co je rozdíl mezi AVL a ČČ stromy?

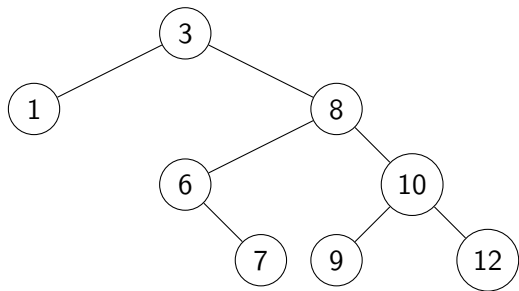
# Cerveno-černý strom vs AVL

- ▶ AVL jsou vyrovnanější.
- ▶ Ale to je díky více úpravám při vkládání/odstraňování uzlů.

Viz =Geeks=

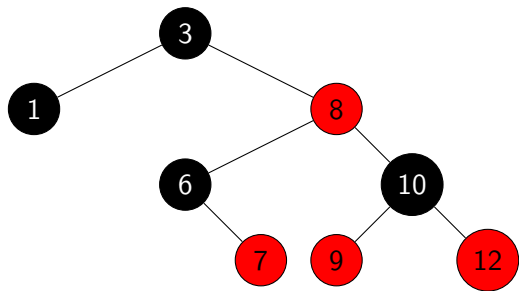


## Cerveno-cerny strom vs AVL



Co je tento strom?

## Cerveno-cerny strom vs AVL



Co je tento strom?

# Vlastnosti

- ▶ Inorder průchod = seřazená čísla.
- ▶ BVS můžeme stvořit jak pomocí preorder/postorder přístupem.

