

STROMY

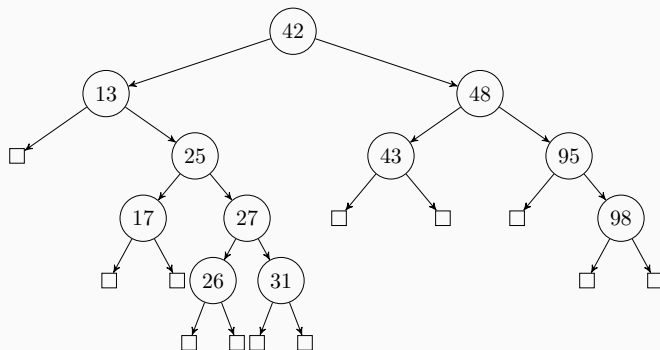
Karel Horák, Petr Ryšavý

9. března 2016

Katedra počítačů, FEL, ČVUT

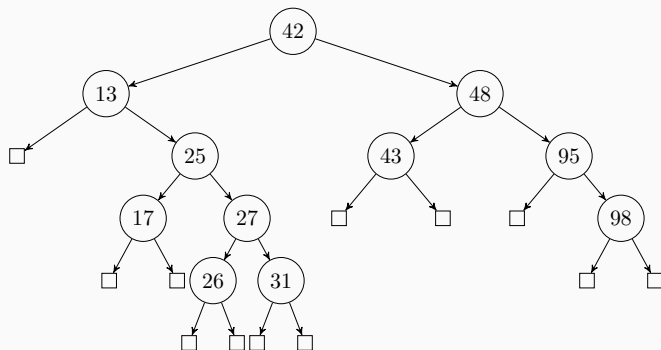
Příklad 1

Uvažme binární strom



Napište kód, který najde nejmenší hodnotu ve stromu, spočte součet všech hodnot ve stromu a určí počet uzlů ve stromu.

Uvažme binární strom

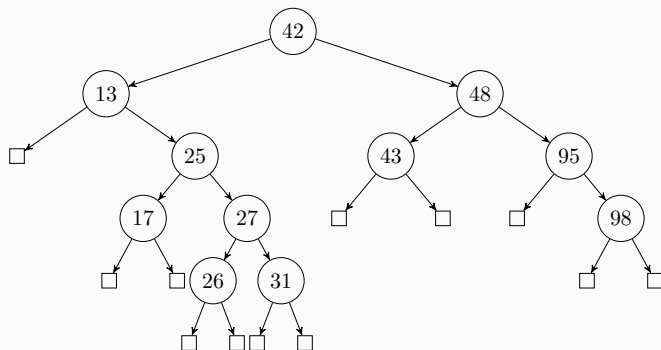


Jaký je průchod stromem inorder? Napište kód, který projde strom inorder.

Průchod stromem inorder je: 13, 17, 25, 26, 27, 31, 42, 43, 48, 95, 98.

Příklad 3

Uvažme binární strom



Jaký je průchod stromem preorder. Napište kód, který projde strom preorder.

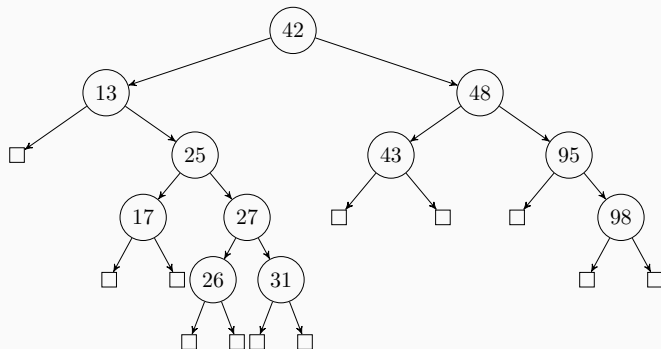
Popište tvar binárního stromu, pro nějž platí, že průchod v pořadí inorder a preorder vytvoří stejnou posloupnost uzlů. (klíče jsou různé)

Průchod stromem preorder je: 42, 13, 25, 17, 27, 26, 31, 48, 43, 95, 98.

Při průchodu preorder je vždy kořen před svým levým a pravým podstromem. Při průchodu inorder následuje po svém levém podstromu. Pokud preorder a inorder dávají stejný výsledek, musí být tedy nutně levý podstrom prázdný. To musí platit pro každý vrchol stromu.

Příklad 4

Uvažme binární strom



Jaký je průchod stromem postorder. Napište kód, který projde strom postorder.

Popište tvar binárního stromu, pro nějž platí, že průchod v pořadí preorder a postorder vytvoří stejnou posloupnost uzlů. (klíče jsou různé)

Průchod stromem postorder je: 17, 26, 31, 27, 25, 13, 43, 98, 95, 48, 42.

V pořadí preorder je kořen stromu první, v pořadí postorder poslední.

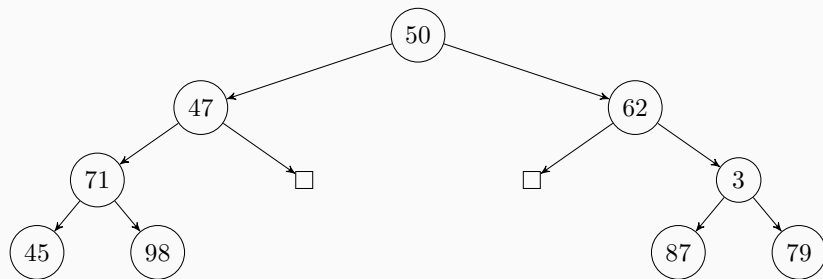
Pokud oba průchody mají být stejné, pak je nutně kořen jediný uzel ve stromu.

Při průchodu daným stromem pořadí inorder a preorder získáme následující posloupnosti klíčů uložených v jeho jednotlivých uzlech:

Inorder: 45 71 98 47 50 62 87 3 79

Preorder: 50 47 71 45 98 62 3 87 79

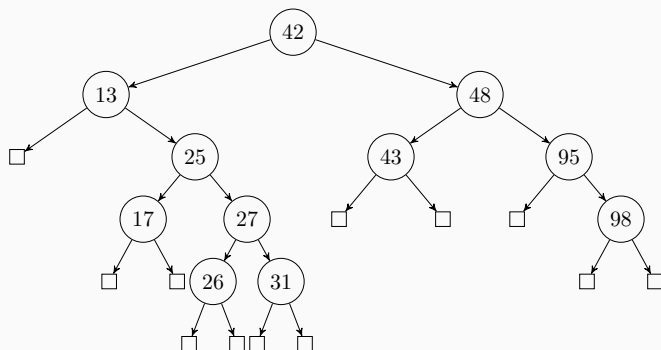
1. Rekonstruuje strom.
2. Navrhněte a formulujte algoritmus, který z uvedených dvou posloupností pro libovolný strom rekonstruuje jeho podobu za předpokladu, že všechny klíče jsou různé.



Obecný algoritmus vezme první navštívený uzel v pořadí preorder jako kořen stromu. Kořen pak najde v průchodu inorder a rozdělí průchody na levý podstrom a pravý podstrom. Rekurzivně je pak aplikovaná stejná myšlenka.

Příklad 6

Uvažme binární strom



Kolik má listů. Napište kód, který spočte počet listů.

Pravidelný (regulární) binární strom má n uzlů. Kolik má listů?

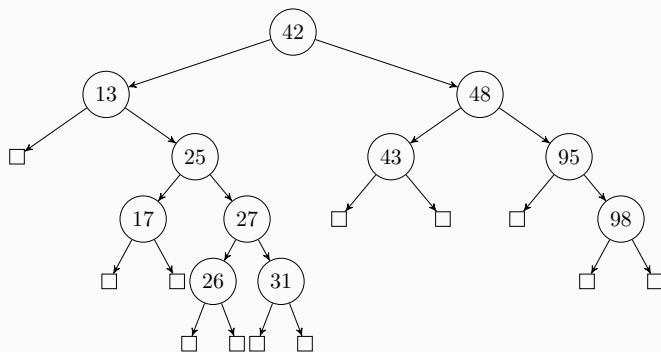
Vyobrazený strom má 5 listů.

Pravidelný strom má o jeden více listů než vnitřních uzlů. Proto počet jeho listů je

$$\frac{n + 1}{2}.$$

Příklad 7

Uvažme binární strom



Jaká je jeho hloubka? Napište kód, který určí hloubku stromu.

Jaká je minimální a maximální možná hloubka binárního stromu s 300 listy? A co v případě ternárního stromu? Jak je to v obecném případě pro n listů a binární strom.

Hloubka vyobrazeného stromu je 4.

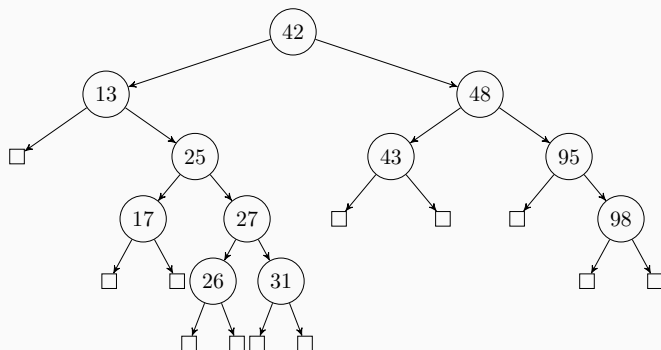
Minimální možná hloubka binárního stromu s 300 listy je $\lceil \log_2 300 \rceil = 9$.

Maximální možná hloubka není omezená.

V případě ternárního stromu je minimální možná hloubka $\lceil \log_3 300 \rceil = 6$ a maximální hloubka není omezená.

Obecně pro n listů je minimální možná hloubka binárního stromu $\lceil \log_2 n \rceil$ a maximální je nekonečno.

Uvažme binární strom



Uvažme operaci `heapify`, na jejímž konci má každý uzel menší hodnotu než všichni jeho přímí potomci. Naimplementujte tuto operaci.