Binární vyhledávací stromy II

doc. Mgr. Jiří Dvorský, Ph.D.

Katedra informatiky
Fakulta elektrotechniky a informatiky
VŠB – TU Ostrava

Prezentace ke dni 12. února 2020



Osnova přednášky

Binární vyhledávací stromy II

Kontrolní otázky

1. Máme za úkol odebrat klíč ze stromu, čili zrušit odpovídající uzel v binárním vyhledávacím stromu. Jak postupujeme, jestliže rušený uzel nemá žádné potomky.
2. Máme za úkol odebrat klíč ze stromu, čili zrušit odpovídající uzel v binárním vyhledávacím stromu. Jak postupujeme, jestliže rušený uzel má jednoho potomka.
3. V binárním vyhledávacím stromu je dán klíč k. Jakým způsobem najdeme, či kde se v tomto stromu nachází nejbližší menší klíč než k?
4. V binárním vyhledávacím stromu je dán klíč k. Jakým způsobem najdeme, či kde se v tomto stromu nachází nejbližší větší klíč než k?
5. Máme za úkol odebrat klíč ze stromu, čili zrušit odpovídající uzel v binárním vyhledávacím stromu. Jak postupujeme, jestliže rušený uzel má dva potomky.
6. Co je to průchod binárním stromem?

Kontrolní otázky (pokrač.)

7. Lze na binární strom aplikovat obecné algoritmy průchodu grafem? Proč?
8. Jakou datovou strukturu byste použili pro průchod stromem po „patrech“?
9. Popište přímý průchod binárním stromem (anglicky preorder).
10. Popište vnitřní průchod binárním stromem (anglicky inorder).
11. Popište zpětný průchod binárním stromem (anglicky postorder).
12. Uveďte příklady využití jednotlivých způsobů průchodu binárním stromem.
13. Uvažujme libovolný způsob průchodu stromem (preorder, inorder, postorder). Z jaké vlastnosti binárního stromu plyne fakt, že si nemusíme uzly ve stromu značkovat (černá, šedá a bílá barva), abychom věděli, který uzel jsme již navštívili a přitom algoritmus skončí, tj. neuvázne v nekonečném cyklu?

Kontrolní otázky (pokrač.)

14. Uvažujme libovolný způsob průchodu stromem (preorder, inorder, postorder). Z jaké vlastnosti binárního stromu plyne fakt, že si nemusíme uzly ve stromu značkovat (černá, šedá a bílá barva), abychom věděli, který uzel jsme již navštívili a přitom žádný uzel nevynecháme?
15. Popište implementaci iterátoru v binárním vyhledávacím stromu. Jak se musíme vypořádat s faktem, že na každý uzel ve stromu mohou navazovat další dva potomci? Jakou musíme použít pomocnou datovou strukturu?
16. Jak najdeme v iterátoru další prvek pro iteraci? Předpokládejme, že iterujeme od nejmenšího prvku směrem k největšímu.
17. Jak poznáme v iterátoru konec iterace?

Kontrolní otázky (pokrač.)

18. Výsledkem algoritmu průchodu grafu do hloubky (DFS) je kostra grafu. DFS algoritmus aplikujeme na binární vyhledávací strom. Jak bude vypadat kostra grafu pro tento druh grafu? Nápověda: Kostra grafu bude shodná s binárním stromem? Nebo bude nějaký uzel či hrana chybět?
19. Mohou existovat při DFS průchodu binárním vyhledávacím stromem zpětné hrany? Nebo jen dopředné? Vysvětlete.

Děkuji za pozornost