BVS cvičení

2012-03-20

Radek Mařík

CS

- Čísla ze zadané posloupnosti postupně vkládejte do prázdného binárního vyhledávacího stromu (BVS), který nevyvažujte. Jak bude vypadat takto vytvořený BVS?

CB

- a) 14 24 5 13 1 3 22 10 19 11
- b) 10 16 5 17 4 15 3 1 23 13 2 11
- c) 17 4 15 2 5 9 1 12 3 19 16 18

2012-03-20

C

Jegth = 7 1 2/2 1/2 1/2 2/2

CS

- Mějme klíče 1, 2, 3, ..., *n*. Číslo *n* je liché. Nejprve vložíme do BVS všechny sudé klíče v rostoucím pořadí a pak všechny liché klíče také v rostoucím pořadí.

03

- Předpokládejte, že každý klíč v tomto BVS je vyhledáván stejně často a že jiné klíče se v něm nevyhledávají

☐ Jaká je průměrná doba operace Find?

24-1

Radek Mařík

CS

- Rředpokládejte, že každý klíč obsažený v tomto BVS je vyhledáván stejně často a že 50% operací Find vyhledává klíč, který se v tomto BVS nevyskytuje.

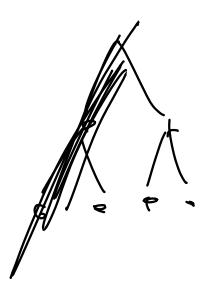


☑ Je dán BVS s *n* uzly. Máme za úkol spočítat hodnotu součtu všech klíčů v tomto stromě.

☐ Jaká bude složitost této operace, když to uděláme efektivně?

03

○ V daném BVS máme smazat pomocí operace Delete všechny listy. Jaká bude asymptotická složitost této akce?





∨ daném BVS máme smazat pomocí operace Delete všechny uzly, které nejsou listy. Jaká bude asymptotická složitost této akce?



- ☑ Uzel binárního binárního vyhledávacího stromu obsahuje tři složky: Klíč a ukazatele na pravého a levého potomka. Navrhněte rekurzívní funkci (vracející bool), která porovná, zda má dvojice stromů stejnou strukturu.
- Dva BVS považujeme za strukturně stejné, pokud se dají nakreslit tak, že po položení na sebe pozorovateli splývají, bez ohledu na to, jaká obsahují data.

03

Napište rekurzivní verze operací: TreeMinimum, která vrátí referenci na uzel s nejmenší hodnotou v BVS.



- Napište funkci, která obrátí pořadí prvků v binárním vyhledávacím stromu.
- Obrácené pořadí znamená, že po výpisu v pořadí inorder (který neimplementujte!), budou prvky srovnány od největšího k nejmenšímu.
- Rroved'te rekurzivně i nerekurzivně.

03

Napište funkci, jejímž vstupem bude ukazatel (=reference) na uzel X v BVS a výstupem ukazatel (=reference) na uzel s nejbližší vyšší hodnotou ve stromu.

03

- Navrhněte algoritmus, který spojí dva BVS *A* a *B*. Spojení proběhne tak, že všechny uzly z *B* budou přesunuty do *A*, přičemž se nebudou vytvářet žádné nové uzly ani se nebudou žádné uzly mazat.
- Přesun proběhne jen manipulací s ukazateli. Předpokládejte, že v každém uzlu v *A* i v *B* je k dispozici ukazatel na rodičovský uzel.

CF

Je dán BVS a klíč x. Máme určit, kolik je v tomto BVS takových klíčů z, pro které platí z < x/2.

Jaká bude asymptotická složitost této operace za předpokladu, že v uzlech stromu neukládáme žádné další pomocné informace?

CF

Ge Je dán BVS a dvojice klíčů x, y (x < y). Máme určit, kolik je v tomto BVS takových klíčů z, pro které platí x < z < y.

Jaká bude asymptotická složitost této operace za předpokladu, že v uzlech stromu neukládáme žádné další pomocné informace?

CS

Operace R z původně vyváženého BVS opakovaně odstraňuje operací Delete vždy ten uzel, který má alespoň jednoho potomka a přitom klíč v odstraňovaném uzlu je největší možný. R končí tesně předtím, než by odstranila kořen BVS.

- A. původní BVS je vyvážený?
- B. původní BVS nemusí být vyvážený?

2012-03-20