BVS
cvičení

1.

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

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

2.

- 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.
- Jaká bude hloubka výsledného stromu?
- Změnil by se nějak tvar stromu, kdybychom lichá čísla vkládali v náhodném pořadí?

3.

- BVS obsahuje 2^n-1 prvků a je vyvážený. Předpokládejte, že během operace Find stráví program v každém navštíveném uzlu právě 1 μs a že další případné činnosti jsou v této době zahrnuty.
- 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?

4.

- BVS obsahuje 2^n-1 prvků a je vyvážený. Předpokládejte, že během operace Find stráví program v každém navštíveném uzlu právě 1 μs a že další případné činnosti jsou v této době zahrnuty.
- Předpokládejte, že každý klíč obsažený v tomto BVS je vyhledáván stejně často a že 50% operací Find vyhledává klíč, který se v tomto BVS nevyskytuje.

- Jaká je průměrná doba operace Find?

5.

- 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?

6.

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

7.

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

8.

- 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.

9.

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

10.

- 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.
- Proveďte rekurzivně i nerekurzivně.

11.

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

12.

- 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.

13.

- 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?

14.

- 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?

15.

- 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.

- Jaká je asymptotická složitost R v případě, že
  A. původní BVS je vyvážený?
  B. původní BVS nemusí být vyvážený?