

Навчальна програма

Інформація про курс

Обговорення

Прогрес

Конспект лекцій

Будь ласка, зверніть увагу! Це завдання на оцінку, яка буде враховуватися для отримання сертифікату.

Для виконання завдання у вас є 2 спроби! Зарахована буде оцінка за останню спробу.

ТЕСТОВЕ ЗАВДАННЯ (5/5 балів)

ПИТАННЯ 1

Задано бінарне дерево з n вузлами та висотою h , яке може бути або не бути бінарним деревом пошуку. Скільки часу необхідно, щоб визначити чи задане дерево задовольняє властивості дерев пошуку?

- ☐ $\Theta(\log n)$
- ☐ $\Theta(h)$
- ☒ $\Theta(n)$ ✓
- ☐ $\Theta(n \log n)$


ПИТАННЯ 2

Нехай є ряд чисел від 1 до 1000, які організовані у вигляді бінарного дерева пошуку і ми виконуємо пошук числа 363. Які з наступних послідовностей **не можуть** бути послідовностями вузлів при пошуку даного елементу?

- ☐ 2, 252, 401, 398, 330, 344, 397, 363
- ☐ 924, 220, 911, 244, 898, 258, 362, 363
- ☒ 925, 202, 911, 240, 912, 245, 363
- ☐ 2, 399, 387, 219, 266, 382, 381, 278, 363
- ☒ 935, 278, 347, 621, 299, 392, 358, 363

ПИТАННЯ 3

Відсортувати множину з n чисел можна наступним чином: спочатку побудувати бінарне дерево пошуку, яке містить ці числа (викликаючи послідовно процедуру `TreeInsert` для вставки чисел в дерево одне за одним), а потім виконати симетричний обхід отриманого дерева за допомогою процедури `InorderTreeWalk`. Чому дорівнює час роботи такого алгоритму в найкращому та найгіршому випадках?

- ☐ Найкращий випадок - $\Theta(n)$; найгірший - $\Theta(n \log n)$
- ☒ Найкращий випадок - $\Theta(n \log n)$; найгірший - $\Theta(n^2)$ 
- ☐ Найкращий випадок - $\Theta(n)$; найгірший - $\Theta(n^2)$
- ☐ Найкращий і найгірший випадки - $\Theta(n \log n)$
- ☐ Найкращий і найгірший випадки - $\Theta(n^2)$

ПИТАННЯ 4

Ключі з однаковими значеннями приводять до появи проблем у роботі бінарних дерев пошуку. Розгляньте три модифікації процедури вставки елементів в дерево `TreeInsert` та оцініть їх час роботи, у випадку коли необхідно вставити послідовно n однакових ключів у початково порожнє дерево (ви повинні дати оцінку часу роботи всієї послідовності викликів процедури `TreeInsert` або її модифікацій, а не якомусь окремому виклику).

1. Оригінальна процедура `TreeInsert`

2. Будемо зберігати в кожному вузлі x деяку логічну змінну $b(x)$. Якщо додається новий вузол із значенням таким самим, як і у вузлі x , то, якщо $b(x) = False$, то елемент буде додано до лівого піддерева, інакше - до правого. Значення цієї змінної у вузлах нащадках x встановлюється протилежним до $b(x)$. Значення $b(x)$ в корені дерева обирається випадково.
3. Так само вводиться додаткова змінна $b(x)$ в кожному вузлі. Тільки після кожного додавання нового елементу з таким самим значенням, що й в x , це значення $b(x)$ змінюється на протилежне.

Дайте оцінку часу роботи описаних процедур при послідовній вставці n однакових елементів в порожнє дерево.

- ☐ 1 - $\Theta(n)$; 2 - $\Theta(n)$; 3 - $\Theta(n)$
- ☐ 1 - $\Theta(n \log n)$; 2 - $\Theta(n)$; 3 - $\Theta(n)$
- ☐ 1 - $\Theta(n^2)$; 2 - $\Theta(n \log n)$; 3 - $\Theta(n)$
- ☐ 1 - $\Theta(n^2)$; 2 - $\Theta(n \log n)$; 3 - $\Theta(n \log n)$
- ☒ 1 - $\Theta(n^2)$; 2 - $\Theta(n^2)$; 3 - $\Theta(n \log n)$ ✓

ПИТАННЯ 5

Означимо загальну довжину шляхів $P(T)$ бінарного дерева T , як суму глибин всіх його вузлів $x \in T$. Позначимо також через T_L та T_R відповідно ліве та праве піддерева дерева T . Яка з наступних рівностей для $P(T)$ є вірною, якщо дерево T містить n вузлів?

- ☐ $P(T) = P(T_L) + P(T_R)$
- ☐ $P(T) = P(T_L) + P(T_R) + n$
- ☒ $P(T) = P(T_L) + P(T_R) + n - 1$ ✓
- ☐ $P(T) = P(T_L) + P(T_R) + \log_2 n$

Остаточна перевірка

Зберегти

Показати відповідь

Ви використали 1 з 2 можливостей надіслати свої матеріали на розгляд.



[Про нас](#) [Преса](#) [FAQ](#) [Контакти](#)

© 2015 Prometheus, some rights reserved

- [Умови надання послуг](#) та [Кодекс Честі](#)

