Київський національний університет імені Тараса Шевченка

Факультет комп'ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №2

«Дерево порядкової статистики»

Виконав студент другого курсу

Групи К-29

Циганов Аркадій Олексійович

2021

**Завдання**

Реалізувати дерево порядкової статистики для класів об’єктів обраної предметної області.

Варіант 17

Предметна область: Кафедра Університета

Об'єкти: Преподаватели, Дисциплины

Примітки: На  кафедре  имеется    множество   
преподавателей.  Для  каждого   
преподавателя  задано  множество   
дисциплин.

**Алгоритм:**

Задача полягає у побудові червоно-чорного дерева для заданих ключів таким чином, щоб за час О(ln(n)) знайти порядкову стататистику і за той самий час знайти ранг елемента.  
Розглянемо основні поняття дерева порядкової статистики та операції над ним.  
  
По-перше, в даному випадку маємо справу зі сферою Аеропортів, тому будувати дерево будемо з об’єктів авіакомпаній. Ключами у даному випадку будуть виступати назви авіакомпаній, інтерпретовані як числа. В іншому ж, дерево будується за загальним зразком, тому розглянемо найбільш загальний алгоритм.  
  
Дерево порядкової статистики T(order-statistictree) являє собою просто червоно-чорне дерево з додатковою інформацією, що зберігається у кожному вузлі. Вузол зберігає такі атрибути: x.key, x.color, x.p, x.left, x.right, x.size. x.size відрізняє структуру вузла дерева порядкової статистики від звичайного червоно-чорного дерева, він зберігає кількість внутрішніх вузлів в піддереві з кореневим вузлом х (включаючи сам х), тобто розмір піддерева. Якщо ми визначимо розмір обмежувача як 0, тобто T.nil.size = 0, то отримаємо тотожність  
x.size = x.left.size + x.right.size + 1.  
У дереві порядкової статистики умова різності всіх ключів не ставиться. Якщо у дереві наявні однакові ключі, визначення рангу виявляється неточним, тому ми усуваємо цю неоднозначність дерева порядкової статистики, визначаючи ранг елемента як позицію, у якій буде виведений даний елемент при центрованому (симетричному) обході дерева.

і-та порядкова статистика множини з n елементів – елемент з і-м в порядку зростання ключем.  
Ранг елемента – його порядковий номер в лінійно впорядкованій множині.

Розглянемо операцію пошуку елемента з заданим рангом: *OS-Select(x,i).* Вона повертає вказівник на вузол, який зберігає і-тий в порядку зростання ключ в піддереві, ключом якого є х.   
Псевдокод має вигляд



Розглянемо процедуру OS-Rank визначення рангу елемента. Дана функція повертає по заданому вказівнику на вузол х дерева порядкової статистики Т позицію даного вузла при центрованому обході дерева.



Правильне визначення рангу та обчислення порядкової статистики виконується за рахунок підтримки висоти дерева. Підтримка висоти працює таким чином як і в ЧЧ-дереві, тобто за рахунок поворотів.



Відновлення розмірів піддерев в процесі поворотів. Зв’язок, навколо якого реалізовується поворот, з’єднує два вузли, атрибути *size*, яких слід обновити.Обновлення локальні, потребуюь інформації з атрибутів *size* вузлів *x, y* та коренів піддерев, вказаних у вигляді трикутників.

Видалення вузла з червоно-чорного дерева також представляє собою двохфазовий процес – перша фаза видаляє вузел із дерева пошуку, що лежить в основі червоно-чорного дерева, а інша – відновляє червоно-чорні властивості, виконуючи не більше трьох поворотів і не вносячи ніяких інших структурних змін.

**Складність: О(ln(n))**

**Мова програмування: C++**

**Основні модулі програми:**

class Teacher – клас викладачів

class Subject – клас предметів

vector<Teacher> Teacher::generatingCompanies(int count) – функція створення множини викладачів

class Tree – модель дерева порядкової статистики з методом поворотів, побудови дерева, вставки, пошуку та видалення вузла.

**Інтерфейс програми:**

Консольний інтерфейс

**Вхідні дані:**Зарезервований набір авіакомпаній. Відображаються у консольному режимі.

**Вихідні дані:**

Приклад видалення вузлів з дерева. (Файл output.txt)

**Висновки:**

В загальному випадку складність алгоритму буде становити **O(nlogn)**

**Література:**

1. Лекція з «Алгоритмів та складності» №3.

2. Т.Кормен «Алгоритмы. Построение и анализ». Часть ІІІ. Глава 14. «Расширенные структуры данных».