

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 2-го курсу
Групи ІПС-21
Ляшенко Матвій Олексійович

Завдання

Реалізувати дерево порядкової статистики на основі червоно-чорного дерева.

Теорія і Алгоритм

Дерево порядкової статистики — це структура даних, яка дозволяє виконувати основні операції (вставка, пошук, отримання k -го елемента) за час, максимально наближений до логарифмічного. Така ефективність досягається завдяки балансуванню — наприклад, використовуючи **червоно-чорне дерево** як базову реалізацію.

Основна ідея алгоритму пошуку k -го елемента полягає у використанні додаткового поля **size**, яке зберігає розмір піддерева для кожного вузла. Нехай **root** — поточний вузол. Якщо:

- **nLeft** — кількість елементів у лівому піддереві **root**,
- тоді якщо $k == \mathbf{nLeft} + 1$ — шуканий елемент знаходиться в **root**,
- якщо $k \leq \mathbf{nLeft}$ — продовжуємо пошук у лівому піддереві,
- інакше — переходимо в праве піддерево та шукаємо $(k - \mathbf{nLeft} - 1)$ -й елемент там, оскільки вже відомо, що ліве піддерево та сам **root** містять $\mathbf{nLeft} + 1$ елементів, менших за шуканий.

У моєму варіанті реалізації значенням кожного вузла є **комплексне число з цілими компонентами**, а порядок визначається за модулем, при рівності — за дійсною частиною.

Предметна область

Варіант №15; Тип даних T5

Тип даних: Комплексні числа з цілими компонентами.

Порівняння: спочатку за модулем, потім — за дійсною частиною.

Складність алгоритму

Червоно-чорні дерева — різновид збалансованих дерев, в яких за допомогою спеціальних трансформацій гарантується, що висота дерева h не буде перевищувати $O(\log n)$. Зважаючи на те, що час виконання основних операцій на бінарних деревах (пошук, видалення, додавання елемента) є $O(h)$, ці структури даних на практиці є набагато ефективнішими, аніж звичайні бінарні дерева пошуку.

Мова програмування

C++

Модулі програми

Лабораторна робота реалізована в одному файлі `main.cpp`, який містить визначення наступних структур і класів:

1. Функція `main()` - Є точкою входу в програму. В ній реалізовано консольний інтерфейс для користувача.
2. Клас `ComplexInt` - Описує комплексні числа з цілими компонентами: дійсна (`real`) та уявна (`imag`) частини. Містить методи:
 - `modulus()` — обчислення модуля комплексного числа
 - оператори `==`, `<` — для порівняння чисел згідно умови варіанту (модуль, потім дійсна частина)
 - `operator<<` — форматоване виведення числа у консоль
3. Структура вузла `Node` - Вузол червоно-чорного дерева. Містить:
 - поле `value` — комплексне число типу `ComplexInt`
 - вказівники на дітей (`left`, `right`) та батьківський вузол (`parent`)
 - поле `color` — колір вузла ('R' або 'B')
 - поле `size` — кількість вузлів у піддереві, що має коренем цей вузол
4. Клас `OrderStatisticRBTree` - Реалізує червоно-чорне дерево з підтримкою порядкової статистики. Основні методи:
 - `insert(val)` — вставка нового елемента з балансуванням
 - `insertFixup(node)` — підтримання властивостей ЧЧ-дерева після вставки
 - `leftRotate(node)` / `rightRotate(node)` — обертання вузлів для балансування
 - `updateSize(node)` / `updateSizeUp(node)` — підтримка актуального розміру піддерев
 - `printTree(...)` — рекурсивний вивід дерева у вигляді гілок
 - `getKth(k)` — повертає вказівник на k-ий за порядком елемент у дереві

Інтерфейс користувача

Користувач вводить дані в консоль. Дані виводяться в консоль.

Тестові приклади

Input sequence	Output sequence
4+2i, 1+3i, 0+0i, 3+4i, -2+1i	<pre>graph TD Root["[B] 1+3i (size=5)"] L1["[B] 0+0i (size=2)"] R1["[R] -2+1i (size=1)"] L2["[B] 4+2i (size=2)"] R2["[R] 3+4i (size=1)"] Root --- L1 Root --- R1 L1 --- L2 L1 --- R2</pre>
Input sequence	Output sequence
4+2i, 1+3i, 0+0i, 3+4i, -2+1i Get k-th element 3	Enter k: 3 k-th element: 1+3i

Висновки

Дерево порядкової статистики — це різновид бінарного дерева пошуку, який, крім стандартних операцій вставки, пошуку та видалення, підтримує додаткову функціональність: **отримання k-го за порядком елемента**. Для цього кожен вузол містить спеціальне поле — **розмір піддерева**, що дозволяє ефективно обчислювати позиції елементів.

Ця структура особливо корисна в задачах, де важливий швидкий доступ до елементів у відсортованому порядку, наприклад, при пошуку медіани, ранжуванні або у редакторських буферах (наприклад, реалізація undo/redo).

Література

- <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-046jintroduction-to-algorithms-sma-5503-fall-2005/video-lectures/lecture-10-red-blacktrees-rotations-insertions-deletions/>
- https://en.wikipedia.org/wiki/Order_statistic_tree
- <https://www.geeksforgeeks.org/red-black-tree-set-2-insert/>
- https://en.wikipedia.org/wiki/Red%E2%80%93black_tree