Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Звіт

лабораторної роботи №3

«Розширюване дерево»

з предмету «Алгоритми та складність»

Над роботою працював:

студент 2 курсу

групи К-29

Маханько Ростислав

2019

**Завдання**. Реалізувати розширюване дерево(splay tree).

**Вхідні дані**: У першому рядку файлу з вхідними даними міститься число команд. Далі для кожної команди її назва, кількість гравців та їх імена, кожне значення з нового рядка.

**Вихідні дані(взаємодія з інтерфейсом)**: Існує п’ять типів запитів:

1. Додати ще одну команду. Зчитуємо назву команди, кількість кравців та їх імена.
2. Шукати команду за ім’ям. Програма виводить “YES”, якщо така команда є, та “NO”, якщо немає.
3. Вивести мінімальний елемент у дереві. Програма виведе назву команди першої за алфавітом.
4. Видалення команди за ім’ям. Зчитується назва команди і якщо в дереві є команда з таким ім’ям, то вона видаляється.
5. Вивести дерево на екран.

**Алгоритм розв’язання**: Розширюване дерево є самозбалансованим бінарним деревом пошуку. Основні операції: вставка, пошук і видалення виконуються за O(log(n)) амортизованого часу. Для багатьох послідовностей невипадкових операцій розширюване дерево виконує роботу краще, ніж інші дерева пошуку, навіть якщо специфічна схема послідовності невідома.

Всі нормальні операції на дереві бінарного пошуку поєднуються з однією базовою операцією, яка називається розширенням(splay). Розширення дерева для певного елемента переставляє дерево таким чином, що елемент розміщується в корені дерева. Це можна зробити за допомогою основної операції пошуку - спочатку виконати стандартний пошук бінарного дерева для розглянутого елемента, а потім скористатися поворотами дерев конкретними способами, щоб привести елемент до вершини. Альтернативно, алгоритм зверху вниз може комбінувати пошук і реорганізацію дерева в єдину фазу.

Операція splay(x) має 3 випадки:

1. zig

Якщо p — корінь дерева з сином x, то робимо один поворот навколо ребра (x,p), роблячи x корнем дерева. Даний випадок є крайнім и виконується тільки один раз в кінці, якщо початкова глибина x була непарною.

1. zig-zig

Якщо p — корінь дерева, а x та p — обидва ліві чи обидва праві діти, то робимо поворот ребра (p,g), де g - батько p, а потім поворот ребра (x,p).

1. zig-zag

Якщо p — не корінь дерева та x — лівий син, а p — правий, чи навпаки, то робимо поворот навколо ребра (x,p), а потім поворот нового ребра (x,g), де g — колишній батько p.

Дана операція виконується за O(d) часу, де d — довжина шляху від x до корня.

**Використана література**:

[https://uk.wikipedia.org/wiki/Розширюване\_дерево](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D1%88%D0%B8%D1%80%D1%8E%D0%B2%D0%B0%D0%BD%D0%B5_%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE)

<https://neerc.ifmo.ru/wiki/index.php?title=Splay-%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE>

<https://www.geeksforgeeks.org/splay-tree-set-1-insert/>

<https://www.geeksforgeeks.org/splay-tree-set-2-insert-delete/>

<https://www.geeksforgeeks.org/splay-tree-set-3-delete/>