Київський національний університет ім. Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Група К-28

Пономарьов Микита Максимович

**Лабораторна робота №3**

м. Київ, 2020р

**Умова**: Реалізувати розширене дерево поверх файлової системи.

**Реалізація**: Основні операції:

* **Пошук**: виконується як в звичайному двійковому дереві пошуку. При знахо-дженні елементу запускаємо Splay для нього.
* **Вставка**: запускаємо Splay від елементу, що додається, і підвішуємо дерева, що вийшли, за нього.
* **Видалення**: знаходимо елемент в дереві, робимо Splay для нього, робимо поточним деревом Merge його дітей.

Допоміжні операції:

* **Splay**: основна операція дерева. Полягає в переміщенні вершини в корінь за допомогою послідовного виконання трьох, наведених нижче, операцій: **Zig**, **Zig-Zig** і **Zig-Zag**. Позначимо вершину, яку хочемо перемістити в корінь за x, її родича — p, а родича p (якщо існує) — g.
* **Zig**: виконується, коли p є коренем. Дерево повертається по ребру між x та p. Існує лише для розбору крайнього випадку і виконується лише один раз в кінці, коли початкова глибина x була непарна.
* **Zig-Zig**: виконується, коли x і p є лівими (або правими) синами. Дерево повертається по ребру між p та x.
* **Zig-Zag**: виконується, коли x є правим сином, а p — лівим (чи навпаки). Дерево повертається по ребру між p та x, а потім — по ребру між x та g.
* **Merge**: для злиття дерев T1 і T2, в яких всі ключі T1 менше ключів в T2, робимо Splay для максимального елементу T1, тоді біля кореня T1 не буде правого дочірнього елемента. Після цього робимо T2 правим дочірнім елементом T1.
* **Split**: для розділення дерева знайдемо найменший елемент, більший або рівний x і зробимо для нього Splay. Після цього відрізуємо біля кореня лівого дочірнього елемента і повертаємо 2 дерева, що вийшли.