**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інформаційних систем**

**Алгоритми та складність**

**Завдання № 3**

**Звіт**

**Виконала:**

студентка групи К-29

**Ходирева Анна Михайлівна**

**Київ-2018**

# Умова завдання

# Реалізуйте розширюване дерево (Splay tree).

**Опис алгоритму**

**Розширюване дерево** (*Splay tree*) – це збалансоване дерево двійкового пошуку з додатковою властивістю: доступ до нещодавно використаних елементів дуже швидкий. Воно виконує основні операції дерева двійкового пошуку (вставка, пошук, видалення) за **O (log n)**, що не гірше, ніж звичайне дерево ДП.

Усі звичайні операції дерева ДП виконуються разом з базовою операцією **splay(x)**. Вона перетворює дерево таким чином, щоб поточний елемент **x** був розміщений у корені дерева. В основі операції **splay(x)** ті ж операції обертання, що використовуються в багатьох інших збалансованих деревах (red-black tree, AVL-tree, тощо).

При цьому операція **splay(x)** лише сприяє балансуванню дерева завдяки операціям, що лежать в її основі, і не може гарантувати повну збалансованість в певний момент часу.

**Особливості реалізації**

Для операцій вставки та видалення використані додаткові операції **split()** та **merge()**.

**split (T, x)** – розбиває дерево **T** на 2 дерева, одне з яких містить елементи тільки строго більші за х, а друге – всі інші. Повертає вказівники на них. Якщо **х** – елемент, який потрібно вставити, тоді можемо його синам присвоїти значення повернених функцією **split()** вказівників, а x помістити в корінь.

**merge (T1, T2)** – поєднує 2 дерева **T1** і **T2** за умови, що **Т2** містить елементи строго більші за всі елементи **Т1**, в одне і повертає вказівник на нього. Якщо **x** – елемент, який потрібно видалити, то можемо замість нього підставити результат роботи функції **merge(x->right, x->left)** і видалити х.

Розширюване дерево представлено у вигляді класу-шаблону **SplayTree<T>**.

Структура **Node**, що представляє вузол дерева, крім лівого та правого синів, має одне додаткове поле **parent** для збереження даних про попередника та уникнення рекурсії в головних операціях.

Клас **InputHandler** реалізує обмін даними з консоллю. Він містить функцію **getInput(Tree),** яка тестує всі головні функції класу **SplayTree**, візуалізуючи роботу з деревом, і може зберігати результат.