Київський національний університет ім. Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Група К-28

Пономарьов Микита Максимович

**Лабораторна робота №5**

м. Київ, 2020р

**Умова**: Реалізуйте персистентну множину на основі червоно-чорного дерева.

**Реалізація**: Червоно-чорне дерево – балансоване бінарне дерево пошуку, для якого виконуються настпуні умови:

* кожна вершина або червона, або чорна
* корінь дерева — чорний
* кожний лист () — чорний
* якщо вершина червона, обидві її дочірні вершини чорні (інакше, батько червоної вершини — чорний)
* усі прості шляхи від будь-якої вершини до листів мають однакову кількість чорних вершин

Всі операції з червоно чорним деревом аналогічні до операцій з звичайним бінарним деревом пошуку, але після кожної вставки/видалення дерево потребує додаткових змін, які зберігають виконання вищевказаних умов. Це реалізується за допомогою процедур remove\_fix, insert\_fix та поворотів rotate\_left, rotate\_right.

Персистентна множина – множина яка зберігає в собі всі свої попередні стани після змін. Для цього можна було б просто зберігати кожну версію як копію і при змінах в множині, просто створювати нову копію, робити зміни з нею та зберігати. Проте, такий підхід використовує додаткової пам’яті для кожної версії, що дуже неефективно.

Нова ідея – реалізувати персистентну множину на основі дерева, для кожної нової версії зберігати новий корінь та копіювати тільки ті вершини, піддерево яких було змінене. Оскільки за одну операцію вставки/видалення ми змнінюємо 1 елемент, то для створення нової версії дерева нам треба буде копіювати всіх предків зміненої вершини, а таких не більше ніж ( – кількість елементів в множині), оскільки дерево балансоване.

Отже, для реалізації персистентної множини на основі червоно чорного дерева достатньо просто копіювати потрібні елементи під час змін.

**Складність**: Операції вставки, пошуку елемента, видалення та балансування дерева виконуються за часу.

Оскільки ми копіюємо не більше, ніж вершин, то затрати в пам’яті будуть , де – кількість версій, тобто кільксть разів коли ми щось додавали чи видаляли з дерева, – максимальна кількість елементів в множині з цих версій.