Київський національний університет ім. Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Група К-28

Пономарьов Микита Максимович

**Лабораторна робота №6**

м. Київ, 2020р

**Умова**: Реалізуйте B+ дерево.

**Реалізація**: B+ дерево – n-арне збалансоване дерево пошуку порядка t, яке задовільняє наступні умови:

* Кожен вузол містить хоча б одни ключ. Ключі всередині одного вузла впорядковані по зростанню. Корінь містить від 1 до 2t – 1 ключа, будь-який інший вузол містить від t-1 до 2t – 1 ключа (t >= 2 – параметр дерева).
* У листів немає нащадків, у всіх інший вузлів з ключами K1, K2, …, Kn є n синів, при чому:
  + Для першого сина всі ключі лежать на проміжку від мінус нескінченності до K1
  + Для 2<=i<=n ключі i-ого сина лежать на проміжку (K(i-1), Ki)
  + Для останнього сина вся ключі лежать на проміжку від Kn до нескінченності
* Глибина всіх листів однакова
* Лист має вказівник на сусіда, яка дозволяє швидко обходити дерево в порядку зростання ключів, та посилання на дані

Для підтримки властивостей B+ дерева при вставці елементів можливо прийдеться розділити деякі вузли, які містять надто багато елементів, що може спричинити каскадні розділення.

**Основні операції над структурою**:

* **Пошук**: Корінь B⁺-дерева є початковою точкою для всього пошуку. Нехай необхідно знайти значення ключа k в B⁺-дереві. Для цього знайдемо листовий вузол, який містить значення. В кожному вузлі треба визначити в якому піддереві треба шукати значення, а далі просто повторити рекурсивно процедуру, поки вузол не виявиться листом.
* **Додавання**: Для додавання нового ключа чи нового запису треба знайти вузол, в який його треба додати. Якщо вузол повністю не заповнений, то просто додаємо в нього ключ, інакше треба розчепити вузол: створити новий вузол і перемістити половину елементів з попереднього вузла в нього додати найменший ключ з нового вузла і вказівник на нього в батьківський вузол якщо батьківський вузол теж заповнений, то його треба теж розділити і повторювати це поки бітьківський вузол не треба буле розчепляти. Якщо розчеплюється корінь, то треба створити новий корінь, який матиме один ключ та два вказівника (ключ, який ми додали в корінь треба видалити з його попереднього вузла)
* **Видалення**: Для початку треба знайти потрібний вузол (аналогічно до пошуку), видалити, а потім зробити потрібні зміни для підтримки властивостей дерева. Якщо вузол хоч на половину заповнений, то нічого робити не треба. Інакше зробити спробу перерозподілу елементів, тобто додати у вузол елемент з «брата» — елемента з спільним предком. Якщо і цього зробити не можна, то вузол треба з’єднати з «братом». Якщо відбулося об’єднання, то ключ чи запис, які вказують на видалений вузол чо його «брата» треба видалити з батьківського вузла. Об’єднання може бути і з корнем, тоді відбувається зменшення висоти дерева.

**Складність алгоритму**:Складність всіх вищевказаних операцій , де n – кількість елементів в дереві, а t – константа, що характеризує кількість елементів в одному вузлі.