Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Лабораторна робота №7

Виконав: Олексишин Олексій

Група К-28

Кафедра інтелектуальних програмних систем

**2020**

**Предметна область**

Вариант 3

Предметная область Отдел кадров Объекты Подразделения, Сотрудники Примечание Имеется множество подразделений предприятия. В каждом подразделении работает множество сотрудников.

Для того щоб дану структуру даних можна було помістити в дерево я перевантажив operator< (та інші оператори порівняння), для виведення в консоль operator<<.

**Завдання**

Реалізуйте структуру даних біноміальна піраміда.

**Алгоритм і теорія**

Біноміальна піраміда – набір біноміальних дерев різних розмірів. Біноміальне дерево – дерево, висота якого рівна степеню двійкі і яке будутється наступним чином:

1. B0 – просто дерево з одного елемента (кореня)
2. дерево Bn – дерево Bn-1 до якого ще додали дерево Bn-1 лівим сином

Нехай в піраміді має бути k значень. Тоді розклавши число k в двійкову систему числення і переписавши як суму степенів двійки ми отримаємо яких розмірів мають бути відповідні біноміальні дерева. Звісно, для кожного з біноміальних дерев в піраміді також має виконуватися властивість піраміди: кожен елемент не менше за свого батька.

Для підтримки властивостей біноміально піраміди при додаванні та вилученні елементів використовується процедура merge, яка зливаєт дві піраміди в одну.

Merge

Беремо дві піраміди і починаємо зливати їх дерева, починаючи з найменших, якщо вони однакового розміру. Тут можуть бути два варіанти:

* є 2 дерева одного розміру, тоді просто зливаємо їх в одне взівши дерево з більшим кореневим елементом і зробивши інше його правим сином
* якщо після якогось попереднього об’єднання виявилося 3 дерева однакового розміру, тоді просто зливаємо два дерева, а інше залишаємо

Insert

Створюємо нову піраміду з 1 елемента і зливаємо її з нашою пірамідою.

GetMin

Мінімальний елемент – один із коренів біноміальних дерев

ExtractMin

Знаходимо за допомогою GetMinimum мінімальний елемент, видаляємо його. Оскільки його діти – теж біноміальні дерева різних розмірів, то з них просто складаємо ще одну піраміду і зливаємо її з нашою.

DecreaseKey

Для того щоб зменшити елемент, його треба знайти, а потім зменшити. Якщо властивість піраміди тепер не виконується, то треба міняти цей елемент з батьківським, поки він не стане коренем, або буде не менше свого батька.

Delete

Для початку знаходимо елемент, зменшуємо його до дуже малого значення (мінус нескінченність), він стає мінімумом, а потім просто робимо ExtractMin/

**Складність алгоритму**

* Merge: O(log(n))
* Insert: максимальна – O(log(n)), середня – O(1) (амортизована)
* Minimum: O(log(n)), бо в піраміді буде O(log(n)) дерев
* ExtractMin: O(log(n))
* DecreaseKey: O(log(n))
* Delete: O(log(n))

**Реалізація (мова С++)**

Програма реалізує клас bheap, що містить в собі реалізацію інтерфейсу для роботи з біноміальною пірамідою (insert, delete, і тд) та всі інші вказані вище процедури.

**Основні модулі програми**

У файлі bheap.hpp знаходиться реалізація структури даних та потрібні для роботи з ним методи (delete\_key, decrease\_key, ectract\_min, insert, merge, min) Файл struct.hpp містить реалізацію класів предметної області. В файлі main.cpp міститься інтерфейс користувача та приклад роботи з реалізованою структурою даних.

Код можна знайти в моєму репозиторії на github за посиланням <https://github.com/ooleksyshyn/semester4/tree/master/algorithms/lab7>

**Інтерфейс користувача та приклади**

Програма дозволяє взаємодіяти з класом через простий інтерфейс – вставка, видалення, пошук елемента за ключем та. В коді є наведений приклад використання програми.

**Літературні джерела**

* <https://ru.wikipedia.org/wiki/Биномиальная_куча>
* <https://habr.com/ru/post/135232/>
* <https://github.com/ooleksyshyn/semester4/tree/master/algorithms/lab7>