КПІ ім. Ігоря Сікорського

Інститут прикладного системного аналізу

Кафедра Системного проектування

Алгоритми та структури даних

**Лабораторна робота №2.3**

**«Дослідження структур даних пріоритетна черга та купа»**

Виконав:

Студент групи ДА-02

Рудік Андрій

Варіант №16

Мета роботи:

Ознайомитись і дослідити структуру даних бінарна купа. Набути навичок реалізації абстрактної структури даних черга з пріоритетом на основі бінарної купи мовою програмування С++. Ознайомитись з механізмом перевантаження операторів та дослідити особливості сортування купою (пірамідального сортування). Порівняти власну реалізацію пріоритетної черги з готовим бібліотечним рішенням STL.

Завдання

Обрати одну із запропонованих задач та реалізувати пріоритетну чергу для ефективнного вирішення цієї задачі.

1. Створити структуру для зберігання об’єктів з характеристиками відповідно до обраної задачі:

* Обрати певний власний критерій для визначення більш пріоритетного об’єкту на основі його характеристик
* Перевантажити оператор “менше” (operator<) для порівняння двох об’єктів за критерієм
* Перевантажити оператор “дорівнює” (operator==) для визначення рівності двох об’єктів за цим самим критерієм

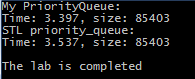
1. Реалізувати пріоритетну чергу на основі незростаючої бінарної купи
   1. Створити структуру PriorityQueue, яка буде містити в собі бінарне дерево для роботи бінарної купи. Для зберігання бінарного дерева можна використати звичайний динамічний масив, що вже був реалізований у минулих роботах.
   2. Реалізувати основні методи переміщення по бінарному дереву:
      1. getParent(index) знайти індекс батьківського вузла
      2. getLeftChild(index) знайти індекс лівого дочірнього вузла
      3. getRightChild(index) знайти індекс правого дочірнього вузла
   3. Реалізувати внутрішні методи для підтримки властивості незростаючої (max-heap) купи:
      1. siftUp(index) просіяти елемент вверх по дереву
      2. siftDown(index) просіяти елемент вниз по дереву
   4. Реалізувати основні методи для роботи пріоритетної черги:
      1. push(object) додати новий елемент в чергу
      2. top() отримати верхній елемент з черги
      3. pop() видалити верхній елемент з черги
      4. size() знайти кількість елементів в черзі
      5. empty() перевірити чергу на пустоту
2. Провести тестування, використавши вказану нижче функцію testPriorityQueue(). Перевірити правильність та швидкість роботи, порівнявши з готовим бібліотечним рішенням STL priority\_queue.
3. Продумати реалізацію сортування купою (пірамідального сортування) на базі бінарної купи, вміти обгрунтовано пояснити та показувати в коді.

Лістинг

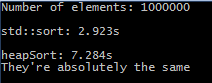
[GitHub](https://github.com/franmajka/algorithms-and-data-structures/tree/master/laba_2-3/laba_2-3)

Результати

Тест черги з пріоритетами:



Порівняння heap sort із стандартною функцію сортування



Висновок

Я ознайомився і дослідив структуру даних бінарна купа. Набув навичок реалізації абстрактної структури даних черга з пріоритетом на основі бінарної купи мовою програмування С++. Ознайомився з механізмом перевантаження операторів та дослідити особливості сортування купою (пірамідального сортування). Порівняв власну реалізацію пріоритетної черги з готовим бібліотечним рішенням STL.