**Міністерство освіти і науки України**

**Львівський національний університет імені Івана Франка**

**Факультет прикладної математики та інформатики**

Кафедра програмування

Лабораторна робота №7

**Черга та Черга з пріоритетом**

з курсу ‘‘Алгоритми та структури даних’’

Виконав:

Студент групи ПМІ-12

Бенько Володимир Сергійович

Львів – 2023

**Черга**

Черга – лінійна, динамічна структура даних, що працює за принципом «перший прийшов — перший пішов» (FIFO — first in, first out). У черги є голова та хвіст. Елемент, що додається до черги, опиняється в її хвості. Елемент, що видаляється з черги, знаходиться в її голові.

Реалізована списком черга містить приватну структуру Item, яка містить значення та вказівник на наступний елемент черги, вказівники на голову та хвіст, довжину черги та набір публічних методів:

* IsEmpty() - повертає істину, якщо черга порожня
* GetSize() - повертає довжину черги
* Head() - повертає значення з голови черги, кидає underflow\_exeption якщо черга порожня
* Enqueue(T val) - додає елемент в кінець (хвіст) черги
* Dequeue() - видаляє елемент з початку (голови) черги, повертає результат в вигляді логічного значення
* Clear() - видаляє всі елементи з черги

**Складність роботи методів:**

* IsEmpty: O(1)
* GetSize: O(1)
* Head: O(1)
* Enqueue: O(1)
* Dequeue: O(1)
* Clear: O(n)

**Просторова складність черги:** O(n)

**Черга з пріоритетом**

Черга з пріоритетами — це нелінійна, динамічна структура даних, що призначена для обслуговування множини елементів, кожний з яких додатково має "пріоритет", пов'язаний з ним. У пріоритетній черзі першим обслуговується елемент, який має найвищий пріоритет, відповідно елемент, що має найнижчий пріоритет буде обслугований останнім.

Реалізована деревом черга з пріоритетом містить приватну структуру Node, яка містить значення вершини, посилання на правого та лівого синів та методи для обходу дерева; також вона містить вказівник на корінь дерева, лічильник елементів в черзі та набір публічних методів:

* IsEmpty() - повертає істину, якщо черга порожня
* GetCount () - повертає кількість елементів в черзі
* PeakMin() - повертає значення з найменшим пріоритетом в черзі, кидає underflow\_exeption якщо черга порожня
* PopMin() - видаляє елемент з найменшим пріоритетом з черги, повертає результат в вигляді логічного значення
* PeakMax() - повертає значення з найбільшим пріоритетом в черзі, кидає underflow\_exeption якщо черга порожня
* PopMax() - видаляє елемент з найбільшим пріоритетом з черги, повертає результат в вигляді логічного значення
* Enqueue(T val) - додає елемент в чергу
* Clear() - видаляє всі елементи з черги

**Складність роботи методів:**

* IsEmpty: O(1)
* GetSize: O(1)
* PeakMin: O(log₂ n)
* PopMin: O(log₂ n)
* PeakMax: O(log₂ n)
* PopMax: O(log₂ n)
* Enqueue: O(log₂ n)
* Clear: O(n)

**Просторова складність черги з пріоритетом:** O(n)

**Приклад:**

Щоб переконатись, що всі функції працюють правильно, в програмі написані юніт-тести. Усі вони проходять успішно:

Зображення, що містить текст

Автоматично згенерований опис

**Висновок:**

Черга використовується, коли елементи не мають внутрішнього порядку або коли важливо зберігати послідовність їх додавання, а черга з пріоритетом використовується тоді, коли елементи мають відмінності у своїй важливості або терміновості. Знання про ці структури даних допоможе нам ефективно організувати наші проекти та забезпечити правильну обробку даних в порядку їх важливості.