Динамична опашка с два края

Изготвено от Георги Газепов

Проектът представлява реализация на опашка с два края от стандартната шаблонова библиотека на C++, като се основава на цикличен масив. Реализацията позволява проверка дали опашката е празна, както и добавяне и премахване на елементи от двата края на опашката за линейно или константно време, съответно в зависимост от това дали се налага опашката да се преоразмерява или не.

1. Архитектура

Dequeue

Класът Dequeue съдържа цялата функционалност на проекта. Основните му характеристики са:

- **Private:**
  - ***arr*** – Шаблонен масив, в който се пазят всички елементи на опашката
  - ***size*** – Текущият брой на елементите в опашката.
  - ***capacity*** – Броят елементи, който опашката може да побере преди да е необходимо преоразмеряване.
  - ***startIndex*** – Индексът от масива, който отговаря на начало на опашката.
  - ***endIndex*** – Индексът от масива, който отговаря на края на опашката.
  - ***calledByResize*** – Булева стойност, която служи за флаг, който се вдига, ако е нужно преоразмеряване на опашката.

- ***void CopyFrom(const Dequeue<T>& other)*** – копира всички данни на даден обект. Използва се в копи-конструктора и операторът за присвояване на класа.

- ***void Free()*** - Изтрива опашката и освобождава заетата от нея памет. Използва се в операторът за присвояване и деструкторът на класа.

- ***Resize(const bool downOrUpSize, const bool BackOrFront)*** – Преоразмерява опашката, като я уголемява или намаля спрямо стойността на *downOrUpSize* (уголемява при подадена стойност true и намаля при подадена стойност false). В случай, че има нужда опашката да се уголемява, в зависимост от стойността на *BackOrFront*, функцията оставя (при стойност true) или не оставя (при стойност false) празно поле в началото на масива. Преоразмеряването се извършва за време **Θ(n).**

➢ **Public:**

- ***Dequeue()*** – Конструктор по подразбиране. Задава *size, capacity*, *startIndex, endIndex u calledByResize* съответно със стойности 0, 4, -1, -1, false

- ***Dequeue(const Dequeue<T>& other)*** – Копи-конструкторът на класа. Създава нов обект с всички характеристики на other.

- ***Dequeue& operator=(const Dequeue<T>& other)*** – Операторът за присвояване на класа. Присвоява всички данни от обектът *other* към вече съществуващият текущ обект.

- ***~Dequeue()*** – Деструкторът на класът. Извиква *Free()*, с помощта на която освобождава заетата памет.

- ***bool IsEmpty() const*** – Проверява дали опашката е празна и връща истина, ако е и лъжа в противен случай.

- ***void PushFront(const T el)*** – Добавя шаблонен елемент в началото на опашката. Проверява дали опашката е пълна и извиква функцията *Resize*() ако е. Добавянето става за константно време, ако не се налага опашката да се преоразмерява и за линейно в противен случай.

- ***void PushBack(const T el)*** – Добавя шаблонен елемент в края на опашката. Проверява дали опашката е пълна и извиква функцията

Resize() ако е. Добавянето става за константно време, ако не се налага опашката да се преоразмерява и за линейно в противен случай.

- ***void PopFront()*** – Премахва елементът в началото на опашката. Премахването става за константно време, ако не се налага опашката да се преоразмерява и за линейно в противен случай. Хвърля изключение, ако опашката е празна.

- ***void PopBack()*** - Премахва елементът в края на опашката. Премахването става за константно време, ако не се налага опашката да се преоразмерява и за линейно в противен случай. Хвърля изключение, ако опашката е празна.

- ***T Front() const*** – Връща елементът в началото на опашката. Извършва се за време от **Θ(1).** В случай на празна опашка хвърля изключение.

- ***T Back() const*** – Връща елементът в края на опашката. Извършва се за време от **Θ(1).** В случай на празна опашка хвърля изключение.

- ***int GetStartIndex() const*** – Гетър, който връща индекса на началото на опашката.

- ***int GetEndIndex() const*** – Гетър, който връща индекса на края на опашката.

- ***int GetCapacity() const*** - Гетър, който връща максималният брой елементи, който опашката може да побере, преди да се наложи преоразмеряване.

- ***int GetSize() const*** - Гетър, който връща текущият брой елементи в опашката.

2. Схема на Проекта