# ЗАДАНИЕ

В лабораторной работе необходимо реализовать заданную структуру данных (см. таблицу ниже) и исследовать вычислительную сложность основных операций с ней. Структура данных выбирается в соответствии с вариантом задания, полученным от преподавателя.

|  |  |
| --- | --- |
| **Вариант** | **Структура данных** |
| 1 | **Невозрастающая очередь с приоритетами**  *Операции*: Enqueue, Dequeue, Peek, IncreasePriority, Merge  *Реализация*: биномиальная пирамида |
| 2 | **Неубывающая очередь с приоритетами** (приоритет – целое число, данные – строка)  *Операции*: Enqueue, Dequeue, Peek  *Реализация*: бинарная пирамида |
| 3 | **Невозрастающая очередь с приоритетами** (приоритет – целое число, данные – строка)  *Операции*: Enqueue, Dequeue, Peek  *Реализация*: бинарная пирамида |
| 4 | **Неубывающая очередь с приоритетами**  *Операции*: Enqueue, Dequeue, Peek, DecreasePriority, Merge  *Реализация*: биномиальная пирамида |
| 5 | **Невозрастающая очередь с приоритетами** (приоритет и данные – обобщенные типы)  *Операции*: Enqueue, Dequeue, Peek  *Реализация*: бинарная пирамида |
| 6 | **Неубывающая очередь с приоритетами** (приоритет и данные – обобщенные типы)  *Операции*: Enqueue, Dequeue, Peek  *Реализация*: бинарная пирамида |
| 7 | **Невозрастающая очередь с приоритетами** (приоритет – целое число, данные – обобщенный тип)  *Операции*: Enqueue, Dequeue, Peek, IncreasePriority  *Реализация*: бинарная пирамида |
| 8 | **Неубывающая очередь с приоритетами** (приоритет – целое число, данные – обобщенный тип)  *Операции*: Enqueue, Dequeue, Peek, DescreasePriority  *Реализация*: бинарная пирамида |

|  |  |
| --- | --- |
| **Вариант** | **Структура данных** |
| 9 | **Универсальная очередь с приоритетами** (приоритет и данные – обобщенные типы)  *Операции*: Enqueue, Dequeue, Peek, IncreasePriority, DescreasePriority  *Реализация*: бинарная пирамида |
| 10 | **Невозрастающая очередь с приоритетами**  *Операции*: Enqueue, Dequeue, Peek, IncreasePriority, Merge  *Реализация*: фибоначчиева пирамида |

Порядок работы:

1. исследовать производительность алгоритмов на реализованной структуре данных;

*Указание*: Исследование выполнять аналогично исследованию в лабораторной работе № 1: для 10–20 разных значений размера входных данных провести несколько (4–7) измерений, отбросить максимальные результаты и усреднить.

# ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Сформулируйте общую идею амортизационного анализа.
2. Назовите и опишите основные методы амортизационного анализа.
3. Опишите структуру данных «Динамическая таблица (Dynamic Table)» и покажите, как выполняются операции добавления и удаления элементов.
4. Приведите анализ операций добавления и удаления элементов для структуры данных «Динамическая таблица (Dynamic Table)».
5. Дайте определение абстрактного типа данных «Очередь с приоритетами (Priority Queue)».
6. На основе каких структур данных может быть реализован абстрактный тип данных «Очередь с приоритетами (Priority Queue)»?
7. Покажите, как устроена структура данных «Двоичная пирамида (Binary Heap)». Опишите порядок выполнения операций *SiftUp* и *SiftDown*. Приведите оценку времени их работы.
8. Опишите, как на основе массива построить двоичную пирамиду за время O(*n*).
9. Приведите псевдокод операций *Enqueue*, Dequeue и *Increase*/*DecreaseKey*

для очереди с приоритетами, реализованной на основе двоичной пирамиды.