

Задача 1-1 (20 баллов). Предложите реализацию *очереди* на основе (одного) массива, которая поддерживает операции добавления в конец и удаления из начала. Требуется, чтобы емкость очереди в любой момент времени отличалась от фактического размера не более чем в константу раз, а учетная сложность операций добавления в конец и удаления из начала была константной.

Решение. Также используем **метод банкира** (accounting method).

Для реализации очереди на основе одного массива, которая поддерживает добавление в конец (**enqueue**) и удаление из начала (**dequeue**), используем динамический массив с циклическим буфером и динамическим изменением размера.

Используем два указателя: **front**, указывающий на начало очереди, и **rear**, указывающий на следующий доступный для добавления элемент. Начальная емкость равна 1.

При добавлении (**enqueue**), если массив заполнен, удваиваем его емкость. Элементы копируем в новый массив так, чтобы **front** всегда был на позиции 0. Это требует n операций, где n — текущий размер очереди. Назначаем амортизированную стоимость добавления в 3 единицы, где 1 покрывает само добавление, а 2 идут в "кредит" на удвоение емкости в будущем. Накопленный кредит $2n$ с предыдущих добавлений покрывает затраты на копирование при увеличении размера.

При удалении (**dequeue**), если после удаления размер становится равным четверти емкости, уменьшаем емкость вдвое и копируем элементы, что требует n операций. Амортизированная стоимость удаления также равна 3. При обычном удалении используется 1 единица на перемещение **front**, а 2 идут в "кредит". Эти $2n$ единиц кредита покрывают затраты на уменьшение емкости, когда это необходимо.

Для циклического массива **enqueue** и **dequeue** могут обходить конец массива с помощью операции по модулю. Это позволяет эффективно использовать всю емкость.

Таким образом, амортизированная сложность операций **enqueue** и **dequeue** — $O(1)$, так как кредит с дешевых операций компенсирует редкие дорогостоящие операции изменения размера. Емкость очереди всегда остаётся в константное количество раз больше фактического размера.