Задача – реализовать Дек на кольцевом буфере.

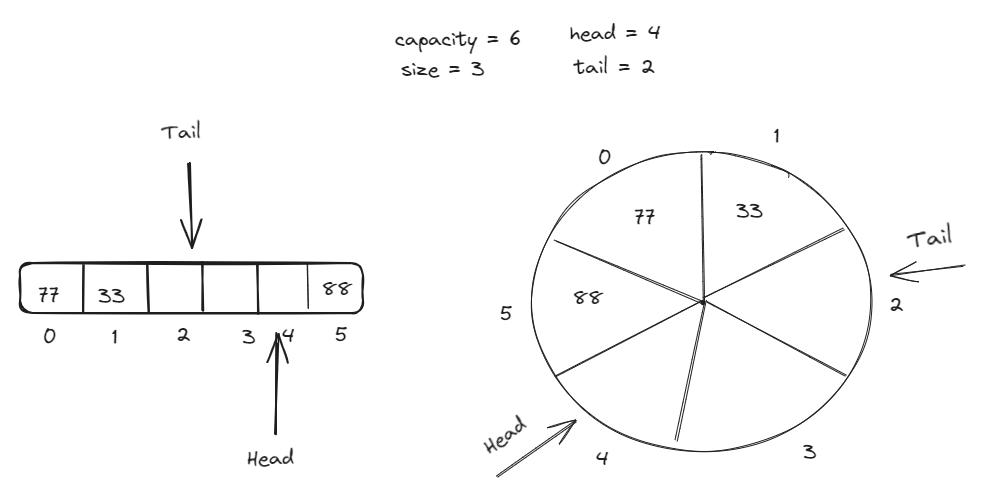
Дек – двусторонняя очередь, в которой элементы могут добавляться и удаляться как из начала, так и с конца очереди. Очередь – структура данных, поддерживающая принцип FIFO.

Для хранения данных выбран массив с настраиваемыми параметрами: capacity – максимальная емкость массива и size – текущее количество элементов в массиве.

Еще две переменные head и tail – хранят индексы текущего значения головы и хвоста Дека.

При добавлении элемента в голову очереди происходит смещение индекса head - против часовой стрелки по кольцу, при добавлении в конец – по часовой стрелке. При удалении элементов из head и tail, смещение индексов происходит наоборот, для head – по часовой, для tail – против.

Рассмотрим схематично массив в некоторый момент времени после наполнения тремя элементам. Например, массив: [77, 33, null, null, null, 88] в котором добавлено 2 элемента в голову и 1 элемент в конец очереди можно представить на картинке:



При добавлении еще одного элемента в голову Дека – будет добавлено значение в ячейку массива с индексом 4, а значение head сместится на одну позицию влево и станет равным 3.

Поскольку известны индексы массива, с которыми предстоит работать, то все операции добавления и удаления выполняются за константное время O(1).

Так как для хранения данных используется массив, то для добавления максимального количества, скажем n элементов, понадобится массив имеющий длину n, поэтому пространственная сложность составляет O(n).