Задача 1-1 (25 баллов). Фиксируем натуральное k. Предложите структуру данных, способную выполнять два действия: `Enqueue`(x) — принять на вход ключ x; `Get-Kth`() — найти k-ю порядковую статистику среди принятых ключей. Учетная сложность `Enqueue` должна быть $O(1)$, сложность `Get-Kth` — $O(k)$ в худшем случае. Совет: группируйте вставки, используйте линейный алгоритм нахождения порядковой статистики.

Решение. Для фиксированного натурального числа k требуется разработать структуру данных, поддерживающую операции `Enqueue`(x) и `Get-Kth`() с заданными ограничениями на временную сложность.

Описание структуры данных:
Будем поддерживать следующие компоненты:

1. Массив S размером не более k, содержащий k наименьших элементов среди всех принятых на данный момент ключей. Массив S поддерживается в отсортированном порядке.
2. Буфер B, представляющий собой неупорядоченный список элементов, поступивших через операции `Enqueue`, но ещё не обработанных.

Операция `Enqueue`(x):

1. Добавить элемент x в буфер B.
2. Если после добавления размер буфера $|B| = k$, выполнить процедуру обработки буфера:
   (a) Объединить множества S и B, получив множество $S \cup B$ размера не более $2k$.
   (b) Найти k наименьших элементов в объединённом множестве $S \cup B$ с помощью линейного алгоритма нахождения порядковой статистики.
   (c) Обновить массив S, сохранив в нём найденные k наименьших элементов в отсортированном порядке.
   (d) Очистить буфер B.

Операция `Get-Kth`():

1. Если буфер B не пуст ($|B| > 0$), выполнить процедуру обработки буфера (аналогично пункту 2 в операции `Enqueue`).
2. После гарантии того, что все элементы обработаны, вернуть максимальный элемент из массива S, который соответствует k-й порядковой статистике среди всех принятых ключей.

Анализ временной сложности:
Операция `Enqueue`(x):

Каждая вставка элемента в буфер B выполняется за $O(1)$. Обработка буфера происходит после каждых k вставок и занимает $O(k)$ времени (объяснение ниже). Таким образом, амортизированная стоимость одной операции `Enqueue` составляет

$$O(1) + \frac{O(k)}{k} = O(1).$$

Операция `Get-Kth`*():*

В худшем случае требуется обработать буфер B, что занимает $O(k)$ времени. После обработки буфера доступ к элементу в массиве S выполняется за $O(1)$, так как массив отсортирован и размер его не более k. Таким образом, общая временная сложность операции `Get-Kth`() составляет $O(k)$.

Обоснование обработки буфера за $O(k)$*:*

- Объединение множеств S и B занимает $O(k)$, так как размеры $|S| \leqslant k$ и $|B| = k$.
- Нахождение k наименьших элементов в множестве размера не более $2k$ можно выполнить за $O(k)$ времени с помощью алгоритма поиска k-й порядковой статистики за линейное время (например, алгоритм медианы медиан).
- Сортировка этих k элементов для поддержки массива S в отсортированном виде занимает $O(k \log k)$, но поскольку k считается константой, то $O(k \log k) = O(k)$.