

**Задача 1-1 (25 баллов).** Пусть задана бинарная куча (min-heap) из  $n$  элементов. Придумайте алгоритм, находящий  $k$  минимальных элементов в ней за время  $O(k \log k)$ . Исходная куча при этом должна остаться без изменений, однако разрешается в процессе работы использовать дополнительную память. Совет: удаление элемента из бинарной кучи состоит в отделении корня и пары поддеревьев.

**Решение.** Будет использоваться дополнительная мин-куча (приоритетную очередь)  $H'$  для хранения потенциальных кандидатов на следующие минимальные элементы. Изначально в  $H'$  помещается корень исходной кучи. На каждом шаге извлекается минимальный элемент из  $H'$ , и его потомки добавляются в  $H'$  как новые кандидаты.

**Описание алгоритма:**

1. Инициализировать пустую мин-кучу  $H'$ .
2. Вставить в  $H'$  корень исходной кучи вместе с его индексом или указателем (для доступа к потомкам).
3. Повторить  $k$  раз:
  - (a) Извлечь минимальный элемент  $(v, p)$  из  $H'$ , где  $v$  — значение узла,  $p$  — позиция узла в исходной куче.
  - (b) Добавить  $v$  в результирующий список минимальных элементов.
  - (c) Если у узла  $p$  есть левый потомок, вставить его в  $H'$ :
    - Вычислить позицию левого потомка  $l = 2p + 1$  (при нумерации с нуля).
    - Если  $l < n$ , вставить  $(v_l, l)$  в  $H'$ , где  $v_l$  — значение в позиции  $l$ .
  - (d) Если у узла  $p$  есть правый потомок, вставить его в  $H'$ :
    - Вычислить позицию правого потомка  $r = 2p + 2$ .
    - Если  $r < n$ , вставить  $(v_r, r)$  в  $H'$ , где  $v_r$  — значение в позиции  $r$ .

**Корректность алгоритма:**

- *Свойство мин-кучи:* В любой момент значение родительского узла не превосходит значений его потомков.
- *Выбор минимальных элементов:* Поскольку мы начинаем с корня (наименьшего элемента) и последовательно рассматриваем потомков извлеченных узлов, мы всегда имеем доступ к следующим по величине элементам.
- *Отсутствие дубликатов:* Каждый узел добавляется в  $H'$  не более одного раза, так как потомки добавляются только при извлечении их родителя.

**Анализ временной сложности:**

- *Количество операций:* За  $k$  итераций мы выполняем  $k$  операций извлечения и не более  $2k$  операций вставки (каждый узел имеет не более двух потомков).

- *Стоимость операций:* Каждая операция вставки и извлечения в мин-куче  $H'$  занимает  $O(\log s)$ , где  $s$  — текущий размер  $H'$ .
- *Оценка размера  $H'$ :* В любой момент времени размер  $H'$  не превышает  $k$ , так как изначально  $|H'| = 1$ , и на каждой итерации мы добавляем не более двух элементов и извлекаем один.
- *Общая временная сложность:*

$$O(k \log k) + O(2k \log k) = O(k \log k).$$

**Сохранение исходной кучи:**

Алгоритм не модифицирует исходную кучу, так как все операции чтения выполняются без изменения структуры данных. Мы используем дополнительную кучу  $H'$  для хранения ссылок на узлы исходной кучи.