

**Задача 1-1 (25 баллов).** Предложите реализацию кучи, позволяющую добавить новый ключ за учетное время  $O(1)$  и извлечь минимальный за время  $O(\log n)$  в худшем случае (здесь  $n$  обозначает текущее число элементов в куче). Совет: группируйте вставки.

**Решение.** Биномиальная куча  $H$  представляет собой коллекцию биномиальных деревьев  $B_k$ , где  $k$  — порядок дерева. Деревья хранятся в порядке возрастания порядков, и для каждого порядка существует не более одного дерева.

**Операции:**

1. *Insert*( $x$ ):

1. Создать новую биномиальную кучу  $H'$  с единственным узлом  $x$ , представляющим собой биномиальное дерево порядка 0.
2. Объединить  $H$  и  $H'$  с помощью операции **Union**, которая сводится к последовательному объединению деревьев одинаковых порядков.

Однако в случае вставки единственного узла операция **Union** упрощается:

- Поскольку  $H'$  содержит только дерево порядка 0, а в  $H$  может быть либо отсутствовать дерево порядка 0, объединение можно выполнить за  $O(1)$  время.

*Амортизированный анализ:*

Поскольку операция **Insert** в биномиальной куче выполняется за  $O(\log n)$  в худшем случае из-за возможного объединения деревьев при совпадении порядков, но амортизированное время вставки составляет  $O(1)$ .

2. *Extract-Min*( $\cdot$ ):

1. Найти дерево в биномиальной куче  $H$ , корень которого содержит минимальный ключ. Это можно сделать за  $O(\log n)$  времени, поскольку количество деревьев в куче не превышает  $\log n$ .
2. Удалить корень минимального дерева  $B_{\min}$  из  $H$ . Остальные деревья остаются в  $H$ .
3. Получить список поддеревьев удаленного корня (его детей), которые сами являются биномиальными деревьями порядков от 0 до  $k - 1$ , где  $k$  — порядок  $B_{\min}$ .
4. Обратить порядок поддеревьев (чтобы сохранить свойства биномиальной кучи) и сформировать новую биномиальную кучу  $H'$  из этих деревьев.
5. Объединить  $H$  и  $H'$  с помощью операции **Union**.

*Временная сложность:*

- Поиск минимального корня выполняется за  $O(\log n)$  времени.
- Удаление корня и обращение списка его детей занимает  $O(\log n)$  времени.

- Объединение двух биномиальных куч  $H$  и  $H'$  занимает  $O(\log n)$  времени, так как количество деревьев ограничено  $\log n$ .
- Итого, операция **Extract-Min** выполняется за  $O(\log n)$  в худшем случае.

#### Амортизированный анализ операций:

- **Insert**: амортизированное время  $O(1)$ .
- **Extract-Min**: время  $O(\log n)$  в худшем случае.

#### Обоснование выбора биномиальной кучи:

Биномиальная куча подходит для данной задачи, так как она обеспечивает:

- Амортизированное время вставки  $O(1)$ , что соответствует требованию учетного времени  $O(1)$  для операции **Insert**.
- Время извлечения минимального элемента  $O(\log n)$  в худшем случае, что соответствует требованию для операции **Extract-Min**.

#### Альтернативный подход с использованием группировки вставок:

Если требуется строгое учетное время  $O(1)$  для вставки, можно использовать следующую стратегию:

1. **Структура данных**: Поддерживать основную кучу  $H$  и буфер  $B$  для новых элементов.
2. **Операция  $\text{Insert}(x)$** :
  - Добавить элемент  $x$  в буфер  $B$ . Это занимает  $O(1)$  времени.
3. **Операция  $\text{Extract-Min}()$** :
  - (a) Если буфер  $B$  не пуст, объединить его с основной кучей  $H$ :
    - Построить кучу  $H_B$  из элементов буфера  $B$ . Это можно сделать за  $O(|B|)$  времени с помощью линейного алгоритма построения кучи.
    - Объединить кучи  $H$  и  $H_B$ . В стандартной бинарной куче объединение не поддерживается эффективно, но можно использовать другую структуру, например двоичную кучу с возможностью объединения.
    - Для эффективного объединения за  $O(\log n)$  времени можно использовать биномиальные или Фибоначчиевы кучи.
  - (b) После объединения буфер  $B$  очищается.
  - (c) Выполнить операцию **Extract-Min** на обновленной куче  $H$ .

#### Анализ временной сложности:

- Вставка в буфер  $B$  выполняется за  $O(1)$  времени.

- При вызове **Extract-Min** объединение буфера с основной кучей может занять  $O(\log n)$  времени.
- Однако, если мы используем биномиальную кучу, объединение двух куч выполняется за  $O(\log n)$  времени, и операция **Extract-Min** также выполняется за  $O(\log n)$  времени.