13. Очередь с приоритетами. Куча. Реализация вставки элемента в кучу.

14. Очередь с приоритетами. Куча. Реализация удаления элемента из кучи.

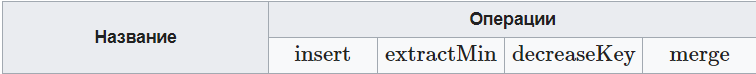
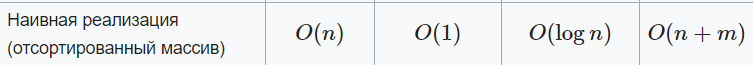
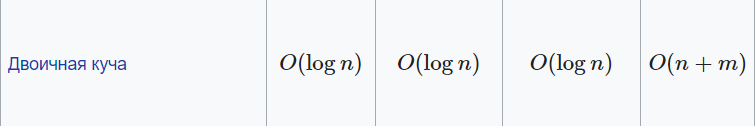
**Очередь с приоритетом**

Очередь с приоритетом — коллекция элементов, где каждый элемент имеет связанный с ним приоритет. Элемент с высшим приоритетом будет обрабатываться раньше, чем элементы с более низким приоритетом.

Очередь с приоритетом можно реализовать различными способами, но обычно главные операции над ними:

1. Вставка элемента с приоритетом — добавление элемента в очередь с учётом его приоритета. В зависимости от реализации, элемент может быть добавлен в начало, в середину очереди или конец.
2. Извлечение элемента с наивысшим приоритетом — удаление элемента из очереди с наивысшим приоритетом. В зависимости от реализации, удаление может происходить из начала, середины очереди или конца.
3. Просмотр элемента с наивысшим приоритетом — просмотр элемента с наивысшим приоритетом без его удаления.
4. Поиск элемента с определенным приоритетом — поиск элемента в очереди с опредёленным приоритетом.

Основные способы реализации очереди с приоритетом включают в себя использование массивов, связанных списков, бинарных куч и древовидных структур. В зависимости от реализации, каждый из этих способов имеет свои преимущества и недостатки в терминах времени выполнения операций.

**Пример**

В качестве примера очереди с приоритетом можно рассмотреть список задач работника. Когда он заканчивает одну задачу, он переходит к очередной — самой приоритетной (ключ будет величиной, обратной приоритету) — то есть выполняет операцию извлечения максимума. Начальник добавляет задачи в список, указывая их приоритет, то есть выполняет операцию добавления элемента.

**Куча**

Двоичная (Max) Куча - полное двоичное дерево, которое поддерживает свойство Max Heap. Двоичная куча - это одна из возможных структур данных для моделирования абстрактного типа данных эффективной очереди с приоритетом.

Куча (Heap) - это структура данных, представляющая собой бинарное дерево, удовлетворяющее основному свойству кучи: приоритет каждого элемента не меньше (или не больше, в зависимости от типа кучи) приоритетов его потомков. В случае максимальной кучи (max-heap) элемент с наивысшим приоритетом располагается в корне дерева, а в случае минимальной кучи (min-heap) - наоборот.

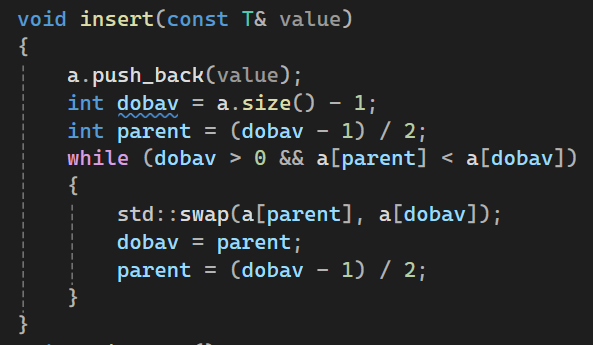
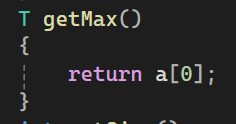
Операция вставки элемента в кучу:

* Добавляем новый элемент в конец массива (или на последнюю позицию бинарного дерева).
* Выполняем "подъем вверх" (Heapify Up), чтобы восстановить свойства кучи. Пока добавленный элемент имеет приоритет выше (для максимальной кучи) или ниже (для минимальной кучи) приоритетов своих родителей, меняем его местами с родителем.

Операция удаления элемента из кучи:

* Удаляем корневой элемент (элемент с наивысшим приоритетом).
* Заменяем корневой элемент последним элементом массива (или дерева).
* Выполняем "опускание вниз" (Heapify Down), чтобы восстановить свойства кучи. Пока корневой элемент имеет приоритет ниже (для максимальной кучи) или выше (для минимальной кучи) приоритетов своих потомков, меняем его местами с наибольшим (для максимальной кучи) или наименьшим (для минимальной кучи) потомком.

Реализация вставки

Реализация удаления



