

Амортизированный анализ

Амортизированный анализ используется в алгоритмах и структурах данных для оценки средней стоимости операций в последовательности, даже если некоторые отдельные операции могут быть дорогими. Примером практического использования амортизированного анализа является динамическая таблица (или динамический массив).

Пример: Динамический массив

Динамический массив — это структура данных, которая позволяет хранить элементы и автоматически изменяет свой размер при добавлении новых элементов.

Описание:

- **Начальный размер:** Начинаем с массива фиксированного размера (например, 4).
- **Добавление элементов:** Когда массив заполняется, создаем новый массив вдвое большего размера и копируем все элементы из старого массива в новый.

Пример использования:

1. Создаем массив размером 4.
2. Добавляем элементы последовательно.
3. Когда массив заполняется, создаем новый массив вдвое большего размера и копируем все элементы.

Амортизированный анализ:

Амортизированный анализ показывает, что даже если операция расширения массива кажется дорогой ($O(n)$ для копирования всех элементов), средняя стоимость добавления элемента остается $O(1)$.

Расчет амортизированной стоимости:

1. **Инициализация:** Начальный массив размера 4, операции добавления элементов.
2. **Добавление элементов:**
 - Добавление 1-го элемента: $O(1)$
 - Добавление 2-го элемента: $O(1)$
 - Добавление 3-го элемента: $O(1)$
 - Добавление 4-го элемента: $O(1)$
3. **Расширение и копирование:**
 - Добавление 5-го элемента: $O(n)$ для копирования (все 4 элемента) + $O(1)$ для добавления
 - Создание нового массива размера 8, копирование всех элементов из старого массива (4 элемента), добавление нового элемента.

Анализ затрат:

- Стоимость добавления элементов: $1 + 1 + 1 + 1 + 5$ (расширение + добавление)
- Общая стоимость: $1 + 1 + 1 + 1 + 1 + 4 = 9$ операций для 5 элементов.

Заключение:

Амортизированный анализ показывает, что несмотря на дорогую операцию расширения массива, средняя стоимость добавления элемента остается $O(1)$. Это делает динамические массивы очень эффективными для многих практических применений, таких как реализация списков или векторов в языках программирования.