

Лабораторная работа: Изменение размера динамического массива

Предмет: Алгоритмы и структуры данных

Выполнила студентка 1 курса

Группа: С3200

ФИ: Анастасия Селиванова

Описание задачи

Рассматривается динамический массив со следующей стратегией изменения размера:

- Изначально выделяется память под N элементов.
- При попытке вставки в полностью заполненный массив ($n = N$), размер увеличивается до $2N$.
- При удалении элемента, если количество элементов становится $n = N/4$, размер уменьшается до $N/2$.

Цель: доказать, что амортизированное время вставки и удаления составляет $O(1)$ методом бухгалтерского учета.

Суть метода

Мы приписываем каждой операции (вставка и удаление) некоторую воображаемую **амортизированную стоимость** в «монетах».

- 1 монета — это реальная стоимость одной записи или удаления в памяти.
- Если мы платим больше 1 монеты, остаток идет в **банк структуры данных**.
- Когда массив нужно расширить или сжать, мы берем накопленные монеты из банка, чтобы оплатить копирование всех элементов в новый массив.

Доказательство для Push (вставка)

Установим стоимость `push` = **3 монеты**.

- **1 монета** тратится сразу на запись нового элемента.
- **1 монета** сохраняется в банке для оплаты будущего перемещения этого элемента при расширении массива.
- **1 монета** откладывается в банк для покрытия копирования других элементов при расширении.

Доказываю, почему этого хватает: Допустим, массив расширился до размера N . В нём было $N/2$ элементов. Чтобы массив снова заполнился и расширился, нужно добавить ещё $N/2$ элементов. За эти вставки мы накопим $N/2 \times 2 = N$ монет в банке. Когда придет время копировать все N элементов в новый массив, у нас будет ровно по 1 монете на каждого. Стоимость копирования полностью оплачена.

Доказательство для Pop (удаление)

Установим стоимость `pop` = **2 монеты**.

- **1 монета** тратится сразу на удаление элемента.
- **1 монета** откладывается в банк для оплаты копирования элементов при сжатии массива.

Доказываю, почему этого хватает: Массив имел емкость N . Сжатие произойдет, когда количество элементов упадет с $N/2$ до $N/4$. То есть нужно совершить $N/4$ операций удаления. Каждая операция откладывает по 1 монете, в итоге накопится $N/4$ монет. Именно столько элементов нужно будет скопировать при сжатии. Стоимость копирования полностью оплачена.

Вывод

Мы использовали метод бухгалтерского учета для амортизированного анализа операций вставки и удаления в динамическом массиве. Каждая операция `push` и `pop` имеет фиксированную амортизированную стоимость в «монетах» (3 для вставки, 2 для удаления), и накопленные монеты полностью покрывают дорогие операции расширения и сжатия массива.

Так как баланс монет никогда не уходит в минус, суммарная амортизированная стоимость любой последовательности из m операций ограничена сверху $O(m)$. Следовательно, амортизированное время выполнения каждой операции равно

$$O(1)$$

и не зависит от размера массива.