Амортизационный анализ. Групповой анализ, метод бухгалтерского учета, метод потенциалов

ceagest

1 Амортизационный анализ

Определение 1.1. *Пусть наша программа состоит из элементарных кусков (операций), i-й из которых работает t_i.*

- *Реальное время – t_i.*
- *Среднее время – $t_{avg} = \frac{\sum_i t_i}{n}$.*
- *Амортизированное время одной операции – $a_i = t_i + \Delta\varphi_i$, где $\Delta\varphi_i$ – некоторая добавка.*

1.1 Групповой анализ

Определение 1.2. *В ходе группового анализа исследователь показывает, что в наихудшем случае суммарное время выполнения последовательности всех n операций равно $T(n)$. Поэтому в наихудшем случае средняя, или амортизированная, стоимость, приходящаяся на одну операцию, определяется соотношением:*

$$\frac{T(n)}{n}$$

Замечание 1.1. *Такая амортизированная стоимость применима ко всем операциям, даже если в последовательности имеется несколько разных их типов.*

Пример (стек с multipop) Пусть есть такой стек (для простоты понимания считаем, что он реализован на односвязном списке. Так не нужно думать о влиянии расширения массива на время операций).

- push(s, x) – добавить элемент.
- pop(s, x) – удалить элемент.
- multipop(s, k) – извлечь k верхних элементов.

Фактическое время работы multipop – $\mathcal{O}(k)$. Однако на практике нас интересует, как будет вести себя стек при выполнении последовательности операций push, pop, multipop.

Замечание 1.2. *Будем считать, что стек реализован на односвязных списках, то есть push работает за $\mathcal{O}(1)$ всегда.*

Анализ стека с multipop

- Заметим, что каждый добавленный элемент в стек можно извлечь не более одного раза.
- Фактически время, необходимое для выполнения последовательности операций push, pop, multipop не превышает $\mathcal{O}(n)$.
- Тогда средняя стоимость операции равна $\mathcal{O}(1)$.
- Заметим, что мы оценили среднее время в худшем случае! То есть в этом методе никак не используются вероятностные рассуждения, а значит так будет всегда.

1.2 Метод бухгалтерского учета

Определение 1.3. *В методе в ходе группового анализа, разные операции оцениваются по-разному, в зависимости от их фактической стоимости. Величина, которая начисляется на операцию, называется амортизированной стоимостью (amortized cost). Если амортизированная стоимость операции превышает ее фактическую стоимость, то соответствующая разность присваивается определенным объектам структуры данных как кредит (credit). Кредит можно использовать впоследствии для компенсирующих выплат на операции, амортизированная стоимость которых меньше их фактической стоимости. Таким образом, можно полагать, что амортизированная стоимость операции состоит из ее фактической стоимости и кредита, который либо накапливается, либо расходуется. Этот метод существенно отличается от группового анализа, в котором все операции характеризуются одинаковой амортизированной стоимостью.*

Замечание 1.3. *К выбору стоимости следует подходить осторожно, ведь мы хотим показать, что стоимость операции невелика. При этом, как и в групповом анализе, нужно, чтобы полная амортизированная стоимость последовательности операций была верхней границей полной фактической стоимости.*

Замечание 1.4. *Для того, чтобы метод имел смысл, необходимо, чтобы полный кредит в любой момент времени был неотрицательным, то есть:*

$$\sum_i a_i \geq \sum_i t_i$$

Пример (анализ стека с multipop)

- Положим амортизированную стоимость операции push равной 2 монетам.
- Добавление элемента в стек будет стоить 1 монету.
- Тогда оставшуюся 1 монету после выполнения push добавляем в credit (для использования впоследствии в качестве оплаты за pop или multipop).
- Заметим, что если амортизированные стоимости операций pop и multipop будут 0 монет, а извлечение будет стоить 1 монету, то credit всегда будет неотрицательным.

Итого, все три амортизированные стоимости равны $\mathcal{O}(1)$. Поскольку полная амортизированная стоимость является верхней границей полной фактической стоимости, то полная фактическая стоимость последовательности операций равна $\mathcal{O}(n)$.

1.3 Метод потенциалов

Определение 1.4. *Пусть S – некоторая структура данных, Ω_S – множество всевозможных состояний этой структуры данных. Функцию $\varphi\colon \Omega_S \to \mathbb{R}$ будем называть потенциалом структуры данных S.*

Определение 1.5. *Пусть дана последовательность операций с соответствующими состояниями структуры данных S: $S_0, S_1, ..., S_n \in \Omega_S$. Тогда амортизированная стоимость i-ой операции:*

$$a_i = t_i + \varphi(S_i) - \varphi(S_{i-1})$$

Замечание 1.5. *Легко видеть, что:*

$$\sum_i a_i = \sum_i (t_i + \varphi(S_i) - \varphi(S_{i-1})) = \sum_i t_i + \varphi(S_n) - \varphi(S_0)$$

Если потенциал можно определить так, что $\varphi(S_n) - \varphi(S_0) \geq 0$, то суммарная амортизированная стоимость даст верхнюю границу полной фактической стоимости. Если полную фактическую стоимость разделить на число операций, то имеет смысл говорить о среднем амортизированном времени.

Замечание 1.6. *На практике не всегда известно, сколько операций может быть выполнено, поэтому, если потребовать $\forall i \in \mathbb{N} \quad \hookrightarrow \quad \varphi(S_i) - \varphi(S_0) \geq 0$, то, как и в методе бухгалтерского учета, будет обеспечена предоплата.*

- *Амортизированные стоимости, определяемые разными потенциалами, могут отличаться.*
- *При этом они все еще будут верхней оценкой.*
- *Если у какой-то операции разность потенциалов положительна – имеем переоценку.*
- *Если меньше – расходуем запасы.*

Пример (анализ стека с multipop) Пусть S – рассматриваемый стек, $\varphi(S_i)$ – число объектов в стеке состояния S_i.

- Тогда $\varphi(S_0) = 0$.
- Так как число объектов всегда неотрицательно, имеем $\forall i \in \mathbb{N} \quad \hookrightarrow \quad \varphi(S_i) - \varphi(S_0) \geq 0$, то есть мы гарантируем предоплату.
- В этом случае полная амортизированная стоимость n операций представляет собой верхнюю границу фактической стоимости.

Вычислим амортизированные стоимости всех операций:

- push: $a_i = t_i + \varphi(S_i) - \varphi(S_{i-1}) = t_i + S_i.size - S_{i-1}.size = t_i + (S_{i-1}.size + 1) - S_{i-1}.size = t_i + 1 = 1 + 1 = 2$ (так как $t_i = 1$ для push).
- pop: $a_i = t_i + \varphi(S_i) - \varphi(S_{i-1}) = t_i + S_i.size - S_{i-1}.size = t_i + (S_{i-1}.size - 1) - S_{i-1}.size = t_i - 1 = 1 - 1 = 0$ (так как $t_i = 1$ для pop).
- multipop: $a_i = t_i + \varphi(S_i) - \varphi(S_{i-1}) = t_i + S_i.size - S_{i-1}.size = t_i + (S_{i-1}.size - k) - S_{i-1}.size = t_i - k = k - k = 0$ (так как $t_i = k$ для multipop).

Все стоимости вышли равными $\mathcal{O}(1)$. Так что полная амортизированная стоимость равна $\mathcal{O}(n)$. Поэтому в наихудшем случае стоимость n операций равна $\mathcal{O}(n)$.