1. Алгоритм: определение, свойства, примеры

1. Сложность алгоритма на конкретном входе

Сложность алгоритма на конкретном входе — это базовое понятие в теории анализа сложности алгоритмов, выражающее число рассматриваемых операций, которое совершает алгоритм, обрабатывая конкретный вход. Сложность на конкретном входе — это всегда число.

3) RAM-машина

Модель машины с одним сумматором, команды программы не могут изменять сами себя. Служит теоретической моделью, в частности, для анализа алгоритмов.

RAM-машина состоит из:

* входной ленты, с которой она может только считывать
* выходной ленты, на которую она может только записывать
* памяти.

Входная лента состоит из последовательности ячеек, в которых записаны целые числа. Каждый раз, когда машина считывает число с входной ленты, головка передвигается на следующую ячейку вправо.

4) Сложность в худшем случае

Сложность в худшем — это функция, зависящая от нормы входа (меры линейного размера входа). Сложность в худшем случае — это максимум затрат, которые можно ожидать от данного алгоритма, если перебрать все возможные входы.

Сложность в худшем определяется как функция

|  |  |
| --- | --- |
| , |  |

5) Асимптотические величины: О-большое, Омега-большое, Тетта-большое

6) Сложность в среднем случае

7) Комбинаторные величины

Конечные множества, на элементы которых могут накладываться определённые ограничения, такие как: различимость или неразличимость элементов, возможность повторения одинаковых элементов и т. п.

8) Оценка средней сложности алгоритма для конечного числа вариантов

9) Оценка средней сложности алгоритма для бесконечного числа вариантов

10) Метод динамического программирования для решения задач: общий принцип, условия применения

11) Метод динамического программирования для решения задач: стратегия разработки решения

12) Жадные алгоритмы решения задач: общий принцип, условия применения

13) Жадные алгоритмы решения задач: стратегия разработки решения

14) Амортизационный анализ. Метод группового анализа

15) Амортизационный анализ. Метод бухгалтерского учета

16) Амортизационный анализ. Метод потенциалов.

Метод потенциалов

|  |
| --- |
| **Теорема (О методе потенциалов)**: |
| Введём для каждого состояния структуры данных величину ΦΦ — потенциал. Изначально потенциал равен Φ0Φ0, а после выполнения ii-й операции — ΦiΦi. Стоимость ii-й операции обозначим ai=ti+Φi−Φi−1ai=ti+Φi−Φi−1. Пусть nn — количество операций, mm — размер структуры данных. Тогда средняя амортизационная стоимость операций a=O(f(n,m)),a=O(f(n,m)), если выполнены два условия:   1. Для любого i:ai=O(f(n,m))i:ai=O(f(n,m)) 2. Для любого i:Φi=O(n⋅f(n,m))i:Φi=O(n⋅f(n,m)) |
| **Доказательство:** |
| ▹▹ |
| a=∑i=1ntin=∑i=1nai+∑i=0n−1Φi−∑i=1nΦin=n⋅O(f(n,m))+Φ0−Φnn=O(f(n,m))a=∑i=1ntin=∑i=1nai+∑i=0n−1Φi−∑i=1nΦin=n⋅O(f(n,m))+Φ0−Φnn=O(f(n,m)) |
| ◃◃ |

**Стек с multipop**[[править](https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BC%D0%BE%D1%80%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7&action=edit&section=8)]

В качестве примера вновь рассмотрим стек с операцией multipop(a)multipop(a). Пусть потенциал — это количество элементов в стеке. Тогда:

1. Амортизационная стоимость операций:
   * apush=1+1=2,apush=1+1=2, так как время выполнения операции pushpush — 11, и изменение потенциала — тоже 11.
   * apop=1−1=0,apop=1−1=0, так как время выполнения операции poppop — 11, а изменение потенциала — −1−1.
   * amultipop=k−k=0,amultipop=k−k=0, так как время выполнения операции multipop(k)multipop(k) — kk, а изменение потенциала — −k−k.
2. Для любого i:Φi=O(n),i:Φi=O(n), так как элементов в стеке не может быть больше nn

Таким образом, f(n,m)=1f(n,m)=1, а значит, средняя амортизационная стоимость операций a=O(1)a=O(1).17) Амортизационный анализ. Сравнение методов.

18) Инвариант цикла. Определение, применение.

Инвариант цикла — логическое выражение, истинное после каждого прохода тела цикла (после выполнения фиксированного оператора) и перед началом выполнения цикла, зависящее от переменных, изменяющихся в теле цикла. Инварианты используются для доказательства правильности результата, полученного циклическим алгоритмом.