

Отчет по лабораторной работе №1

Анализ сложности алгоритмов ($O(1)$ и $O(N)$)

Дата: 2025-10-02

Семестр: 3 курс, 1 полугодие — 5 семестр

Группа: ПИЖ-б-о-23-2(2)

Дисциплина: Анализ сложности алгоритмов

Студент: Мусхажиев Игорь Александрович

Цель работы

Познакомиться с понятием вычислительной сложности алгоритмов на примере константной $O(1)$ и линейной $O(N)$ сложности.

Научиться измерять время выполнения алгоритмов, интерпретировать результаты и строить графики эмпирической сложности.

Теоретическая часть

- **Вычислительная сложность** характеризует рост количества операций алгоритма при увеличении размера входных данных.
- $O(1)$ — константная сложность, время работы не зависит от размера входа.
- $O(N)$ — линейная сложность, время работы растёт пропорционально размеру входных данных.
- Для замеров использовалась функция `timeit.default_timer()` из стандартной библиотеки Python.

Практическая часть

Выполненные задачи

- [x] Реализована функция `is_even(n)` для проверки чётности числа ($O(1)$).
- [x] Реализована функция `find_max(arr)` для поиска максимума в массиве ($O(N)$).
- [x] Написана функция `measure_time()` для измерения времени выполнения.
- [x] Проведены замеры времени на массивах разного размера.
- [x] Построен график зависимости времени выполнения от размера входных данных.

Ключевые фрагменты кода

```
def calculate_sum():
    """Пример функции O(1): сумма двух введенных чисел"""
    a = int(input())
    b = int(input())
    result = a + b
    print(result)
    # Общая сложность: O(1)

def sum_array(arr):
    """Пример функции O(N): сумма всех элементов массива"""
    total = 0
    for num in arr:
        total += num
    return total
    # Общая сложность: O(N)
```

Результаты выполнения

Характеристики ПК

- Процессор: Intel Core i5-12400F
- Оперативная память: 16 GB DDR4
- ОС: Windows 10
- Python: 3.9.13

Результаты замеров времени (мс)

O(N)	sum_array	N=1000	time=0.0345 мс
O(N)	sum_array	N=5000	time=0.1782 мс
O(N)	sum_array	N=10000	time=0.3567 мс
O(N)	sum_array	N=50000	time=1.7834 мс
O(1)	calculate_sum	N=1	time=0.0012 мс (оценка)

Выводы

Функция `calculate_sum` имеет константную сложность $O(1)$ — время работы не зависит от размера входных данных. Функция `sum_array` имеет линейную сложность $O(N)$ — время работы растёт линейно с увеличением N . Экспериментальные данные и график подтверждают теоретические оценки сложности алгоритмов.

Приложения

График зависимости времени от размера входа

[Исходный код программы](#) [Файл с результатами замеров](#) [График производительности \(PNG\)](#)