**Практическая работа 1**

**Решение задачи числа Фибоначчи**

**Цель:** «Изучить практическую реализацию принципа разбиения на подзадачи и рассмотреть оптимизацию временной сложности алгоритма на примере задачи Числа Фибоначчи».

**Задание на практическую работу**

1. Изучить материал в учебнике С.М. Окулова Динамическое программирование Глава 1. Простые задачи. 1.1. Числа Фибоначчи (с. 13 - 16).
2. Реализовать на языке Python 3.\* представленный ниже «наивный алгоритм» реализации задачи числа Фибоначчи.

*«Наивный» алгоритм решения (реализовано на С++)*

int fib0(int n) {

if (n == 0) return 0;

if (n == 1) return 1;

return fib0(n - 1) + fib0(n - 2);

}

Почему, по вашему мнению, алгоритм назван «наивным»?

1. Реализовать на языке Python 3.\* рекурсивный вариант решения задачи из учебника.
2. Реализовать на языке Python 3.\* алгоритм с записью в массив из учебника.
3. Реализовать на языке Python 3.\* упрощённый алгоритм с записью в две переменные.
4. Реализовать на языке Python 3.\* метод вычисления чисел Фибоначчи с временной сложностью с записью соотношений в матричной форме.
5. Для каждой реализации задачи посмотреть скорость решения для выбранных вами 10 чисел в промежутке от начальных значений задачи 0 и 1 до размера int. Сколько времени нужно реализации алгоритма, чтобы вычислить нужное число?

*Примечание:*

Измерение скорости работы функции на Python 3.\* можно сделать импортированием метода библиотеки time

import time

start = time.time()

(your code here)

end = time.time()

print(end - start)

Можно сделать профилированием в среде разработки или др.

# **Требования к отчету по практической работе 1:**

1. Титульный лист;
2. Для каждого варианта решения задачи числа Фибоначчи
   1. решение (алгоритм);
   2. текст программы;
   3. скриншоты выполнения программы;
3. Сравнение скорости реализации алгоритма.
4. Выводы по работе.