

## Лабораторная работа № 7 по курсу дискретного анализа: динамическое программирование

Выполнил студент группы 08-307 МАИ *Боев Савелий*.

### Условие:

Имеется натуральное число  $n$ . За один ход с ним можно произвести следующие действия:

- Вычесть единицу
- Разделить на два
- Разделить на три

При этом стоимость каждой операции – текущее значение  $n$ . Стоимость преобразования - суммарная стоимость всех операций в преобразовании. Вам необходимо с помощью последовательностей указанных операций преобразовать число  $n$  в единицу таким образом, чтобы стоимость преобразования была наименьшей. Делить можно только нацело.

### Метод решения

Функция **main** в этом коде отвечает за минимизацию стоимости преобразования числа  $n$  в 1, применяя последовательно три возможные операции: вычитание единицы, деление на два и деление на три, с учетом стоимости каждой операции.

Сначала, функция считывает число  $n$ , которое нужно преобразовать. После этого, инициализируются массивы для хранения информации о минимальной стоимости (**dp**), применяемой операции (**op**) и предыдущем числе (**prev**).

Цикл в функции проходит через все числа от 2 до  $n$ , применяя каждую возможную операцию и обновляя массивы с информацией о минимальной стоимости, выбранной операции и предыдущем числе на каждом шаге.

После завершения цикла, функция восстанавливает и выводит последовательность операций, которые приводят к минимальной стоимости,

начиная от числа  $n$  и возвращаясь к 1. Также выводится минимальная стоимость преобразования.

## Описание программы

**Инициализация массивов для хранения данных:** Программа начинается с создания и инициализации массивов. Массив **dp** хранит минимальную стоимость преобразования каждого числа до 1, **op** хранит информацию о последней операции, выполненной над каждым числом, и **prev** хранит предыдущее число перед выполнением операции.

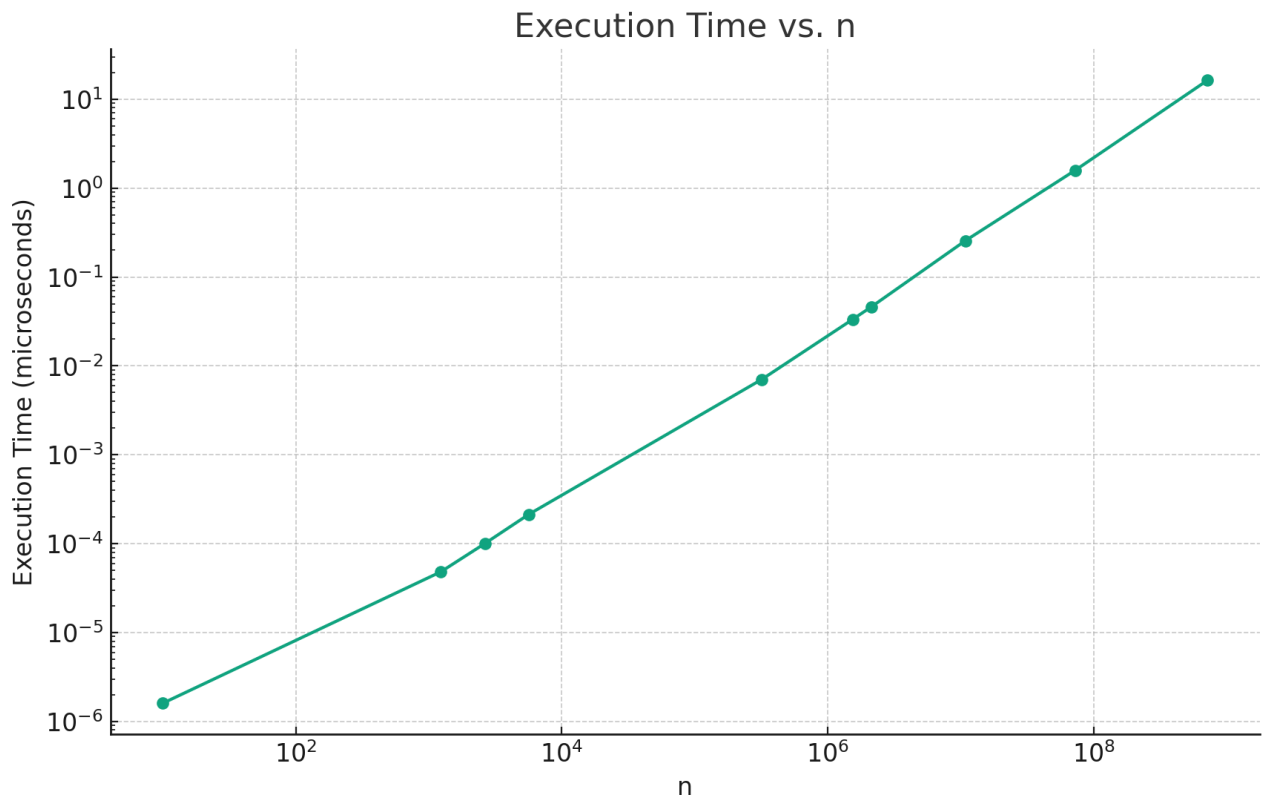
**Определение минимальной стоимости преобразования:** Программа использует динамическое программирование для вычисления минимальной стоимости преобразования каждого числа до 1. Она рассматривает каждое число от 2 до  $n$  и для каждого числа пробует три возможные операции: вычитание единицы, деление на два и деление на три, обновляя минимальную стоимость и сохраняя выполненную операцию и предыдущее число.

**Восстановление последовательности операций:** После вычисления минимальной стоимости преобразования числа  $n$  до 1, программа восстанавливает последовательность выполненных операций, начиная с числа  $n$  и возвращаясь к 1, используя информацию, сохраненную в массивах **op** и **prev**.

**Вывод результатов:** Программа выводит минимальную стоимость преобразования числа  $n$  до 1 и последовательность операций, необходимых для достижения этой минимальной стоимости, предоставляя таким образом полезную информацию о последовательности оптимальных шагов.

.

## Тест производительности



Этот график иллюстрирует зависимость времени выполнения алгоритма от размера входных данных  $n$ . Время выполнения возрастает логарифмически с увеличением  $n$ , что свидетельствует о том, что алгоритм эффективно масштабируется для обработки больших входных данных. Видно, что для больших значений  $n$  время выполнения растет быстрее, что следует учитывать при применении алгоритма на практике.

## **Выводы**

В ходе выполнения лабораторной работы была решена задача методом динамического программирования. То есть для оптимального решения задачи, мы выбирали самое оптимальное решение среди подзадач, а не какой-то одной, как в решениях с помощью жадного алгоритма. Подход динамического программирования применяется в основном в задачах, где нужно искать какой-то минимум или максимум, то есть что-то оптимизировать.