

## • | Сравнительная оценка

### 1. Простота реализации:

- Рекурсивное решение: Код выглядит более лаконично и интуитивно понятен. Рекурсия хорошо описывает саму задачу, что делает код более читабельным.
- Нерекурсивное решение: Код может быть немного более громоздким, но также достаточно прост для понимания.

### 2. Производительность:

- Рекурсивное решение: Может иметь проблемы с производительностью из-за накладных расходов на вызовы функций и может привести к стековому переполнению при больших значениях  $n$ .
- Нерекурсивное решение: Обычно более эффективно, так как избегает накладных расходов на вызовы функций и не имеет риска стекового переполнения.

### 3. Использование памяти:

- Рекурсивное решение: Использует стек для хранения промежуточных вызовов, что может привести к большому потреблению памяти при глубокой рекурсии.
- Нерекурсивное решение: Использует фиксированное количество памяти, так как хранит только переменные для текущего результата и счетчика.

### 4. Удобство отладки:

- Рекурсивное решение: Может быть сложнее отлаживать из-за сложности отслеживания состояния каждого вызова функции.
- Нерекурсивное решение: Обычно проще отлаживать, так как код выполняется в одном контексте без дополнительных уровней стека.