Практическое задание к уроку 3

Урок 3. Множество. Последовательность. Часть 2

Question

1. Как соотносятся понятия “множество” и “последовательность”? (в ответе использовать слова типа: часть, целое, общее, частное, родитель, дочерний субъект и т.д.)

Сначала давайте ознакомимся с терминологией понятий "множества" и "последовательности".

Important

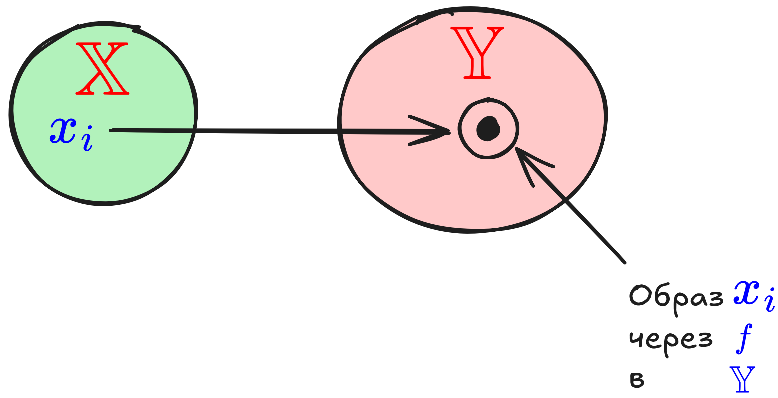
Под множеством мы понимаем объединение в одно целое определенных, вполне различимых объектов нашей интуиции или нашей мысли — так описал понятие "множество" **Георг Кантор**, основатель теории множеств.

Info

Множество — совокупность элементов, обладающих определенными свойствами.

Правило, по которому элементы множества $\mathbb{X}$ связаны с элементами множества $\mathbb{Y}$, называется **функцией** X от Y. X обычно называется **областью определений**, или **доменом**, а Y — областью значений, или **кодоменом**.

Предположим, что x_i — это элемент множества $\mathbb{X}$. Элемент из множества $\mathbb{Y}$, соответствующий x_i благодаря функции f называется образом x_i, отображенным через f во множестве $\mathbb{Y}$.



ⓘ **Info**

Последовательность — функция, областью определения которой является множество натуральных чисел $\mathbb{N}$, а областью значений — множество действительных чисел $\mathbb{R}$ (или другое множество, в зависимости от контекста).

Таким образом, понятие "множество" и "последовательность" соотносятся как **общее** и **частное**.

Множество — это абстрактное понятие, обозначающее совокупность элементов, без учета их порядка и повторений.

Последовательность — это частный случай множества, который характеризуется **упорядоченностью** элементов и **возможностью повторений**.

Другими словами, **последовательность** — это **упорядоченная совокупность элементов**, где каждый элемент имеет свой номер (индекс), определяющий его положение в последовательности.

Примеры:

- **Множество:** {1, 2, 3} — это множество, состоящее из трех элементов: 1, 2 и 3. Порядок элементов не важен, и множество {1, 2, 3} эквивалентно множеству {3, 2, 1}.
- **Последовательность:** (1, 2, 3) — это последовательность, состоящая из трех элементов: 1, 2 и 3. Порядок элементов важен, и последовательность (1, 2, 3) не эквивалентна последовательности (3, 2, 1).

Важно отметить:

- В последовательности могут быть повторяющиеся элементы, например, (1, 2, 2, 3).
- В множестве повторяющиеся элементы не учитываются, например, множество {1, 2, 2, 3} эквивалентно множеству {1, 2, 3}.

Таким образом, **последовательность** — это более **конкретное** и **ограниченное** понятие, чем **множество**, так как она накладывает дополнительные условия на порядок и повторяемость элементов.

> **Question**
>
> Прочитать высказывания математической логики, построить их отрицания и установить истинность.

$$\forall y \in [0;1] : sgn(y) = 1$$

$$\forall n \in \mathbb{N} > 2 : \exists x, y, z \in \mathbb{N} : x^n = y^n + z^n$$

$$\forall x \in \mathbb{R} \exists X \in \mathbb{R} : X > x$$

$$\forall x \in \mathbb{C} \nexists y \in \mathbb{C} : x > y || x < y$$

$$\forall y \in [0; \tfrac{\pi}{2}] \exists \varepsilon > 0 : \sin y < \sin(y + \varepsilon)$$

$$\forall y \in [0; \pi) \exists \varepsilon > 0 : \cos y > \cos(y + \varepsilon)$$

$$\exists x : x \notin \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}\}$$

Последовательность

> **Question**
>
> Даны 4 последовательности. Необходимо:
>
> - исследовать их на монотонность;
> - исследовать на ограниченность;
> - найти пятый по счету член.

$$\{a_n\}_{n=1}^{\infty} = 2^n - n$$

$$\{b_n\}_{n=2}^{\infty} = \frac{1}{1-n}$$

$$\{c_n\}_{n=1}^{\infty} = -1^n + \sqrt{2n}$$

$$\{d_n\}_{n=1}^{\infty} = (-1)^{2n} + \frac{1}{n^2}$$

Последовательность α_n называется монотонной, если для любой пары чисел m и k таких, что $m < k$, выполняется одно из неравенств:

$a_m < a_k$ **монотонно возрастающая последовательность**,

$a_m \leq a_k$ **неубывающая последовательность**,

$a_m \geq a_k$ **невозрастающая последовательность**.

$a_m > a_k$ **монотонно убывающая последовательность**.

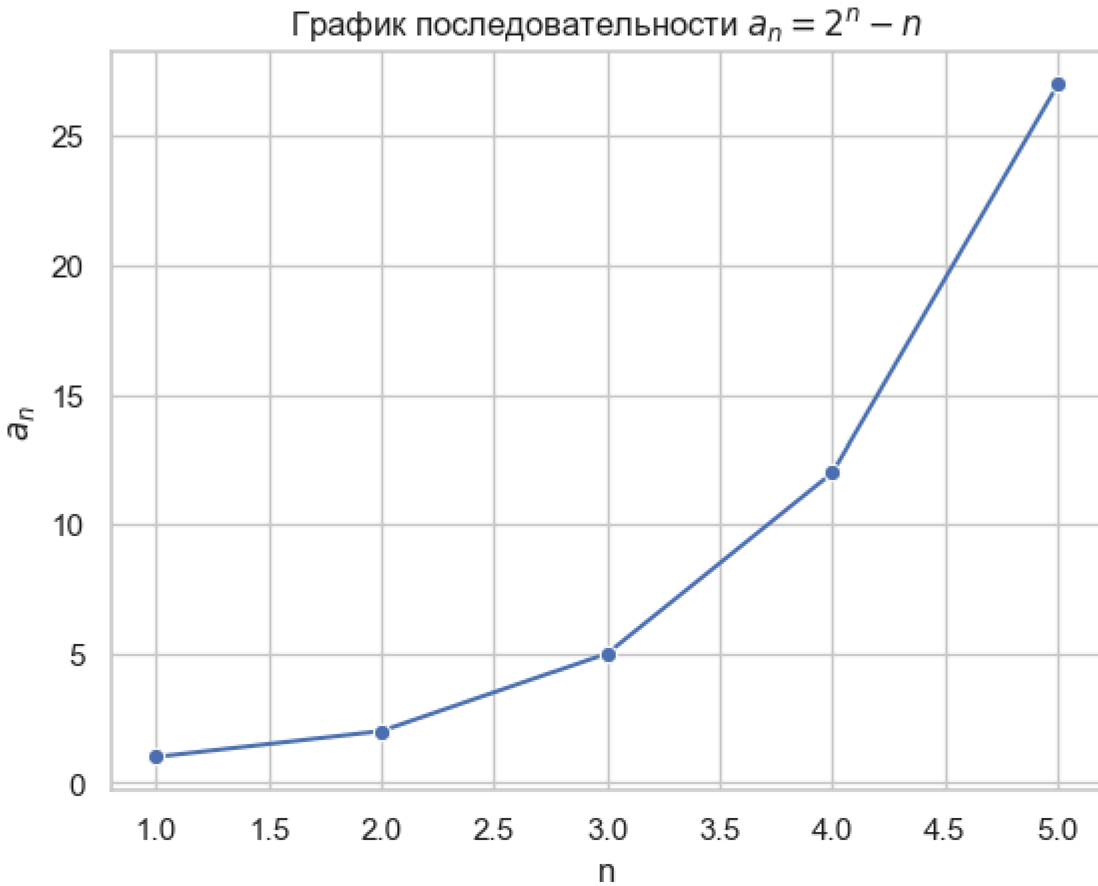
> ✓ **Check**
>
> $\{a_n\}_{n=1}^{\infty} = 2^n - n$

Решение:

Последовательность $\{a_n\}_{n=1}^{\infty} = 2^n - n$ монотонно возрастающая. Убедимся в этом. Сравним два соседних члена a_n и a_{n+1}:

$a_n = 2^n - n$; $a_{n+1} = 2^{n+1} - (n+1)$; $a_{n+1} - a_n$: $a_{n+1} - a_n = 2^{n+1} - (n+1) - 2^n = 2^n - (n+1)$

Следовательно, $a_{n+1} > a_n$, что означает, что последовательность $\{a_n\}$ монотонно возрастает.



Для исследования ограниченности последовательности $\{a_n\}$ необходимо определить, существуют ли такие числа M и m, что для всех $n \geq 1$ выполняется $m \leq a_n \leq M$.

Рассмотрим члены последовательности $a_n = 2^n - n$

При $n \to 1$: $a_1 = 2^1 - 1 = 1$

При $n \to \infty$: $a_n \to \infty$

Таким образом, последовательность $\{a_n\}$ ограничена снизу числом 1.

Нахождение пятого по счету члена

Пятый по счету член последовательности $\{a_n\}$: $a_5 = 2^5 - 5 = 27$

Ответ:

1. Последовательность $\{a_n\}$ монотонно возрастает.
2. Последовательность $\{a_n\}$ ограничена снизу числом 1.
3. Пятый по счету член последовательности равен 27.

✓ Check

$\{b_n\}_{n=2}^{\infty} = \frac{1}{1-n}$

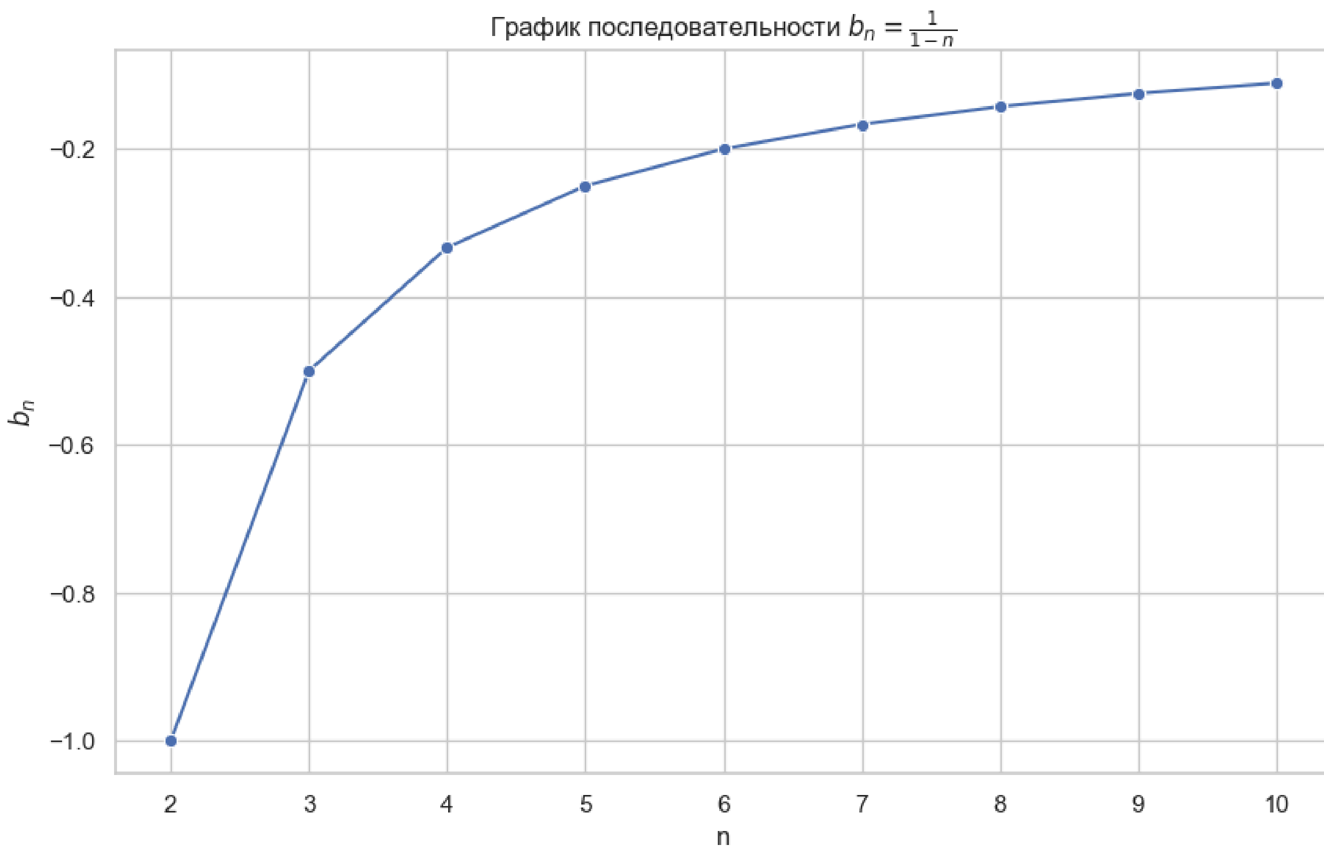
Решение:

Для исследования монотонности последовательности $\{b_n\}$ необходимо сравнить два соседних члена b_n и b_{n+1}:

$b_n = \frac{1}{1-n}$; $b_{n+1} = \frac{1}{1-(n+1)} = \frac{1}{-n}$; $b_{n+1} - b_n$: $b_{n+1} - b_n = \frac{1}{-n} - \frac{1}{1-n}$

Приведем к общему знаменателю: $b_{n+1} - b_n = \frac{1}{-n} - \frac{1}{1-n} = \frac{1-n-(-n)}{-n(1-n)} = \frac{1-n+n}{-n(1-n)} = \frac{1}{-n(1-n)}$

Так как $n \geq 2$, то $-n(1-n) > 0$, следовательно, $\frac{1}{-n(1-n)} > 0$, $b_{n+1} > b_n$, что означает, что последовательность $\{b_n\}$ монотонно возрастает.



Для исследования ограниченности последовательности $\{b_n\}$ необходимо определить, существуют ли такие числа M и m, что для всех $n \geq 2$ выполняется $m \leq b_n \leq M$.

Рассмотрим члены последовательности $b_n = \frac{1}{1-n}$

При $n \to 2$: $b_2 = \frac{1}{1-2} = -1$

При $n \to \infty$: $b_n \to 0$

Таким образом, последовательность $\{b_n\}$ ограничена снизу числом -1 и сверху числом 0.

Нахождение пятого по счету члена

Пятый по счету член последовательности $\{b_n\}$ соответствует $n = 6$ (так как последовательность начинается с $n = 2$): $b_6 = \frac{1}{1-6} = \frac{1}{-5} = -\frac{1}{5}$

Ответ:

1. Последовательность $\{b_n\}$ монотонно возрастает.
2. Последовательность $\{b_n\}$ ограничена снизу числом -1 и сверху числом 0.
3. Пятый по счету член последовательности равен $-\frac{1}{5}$.

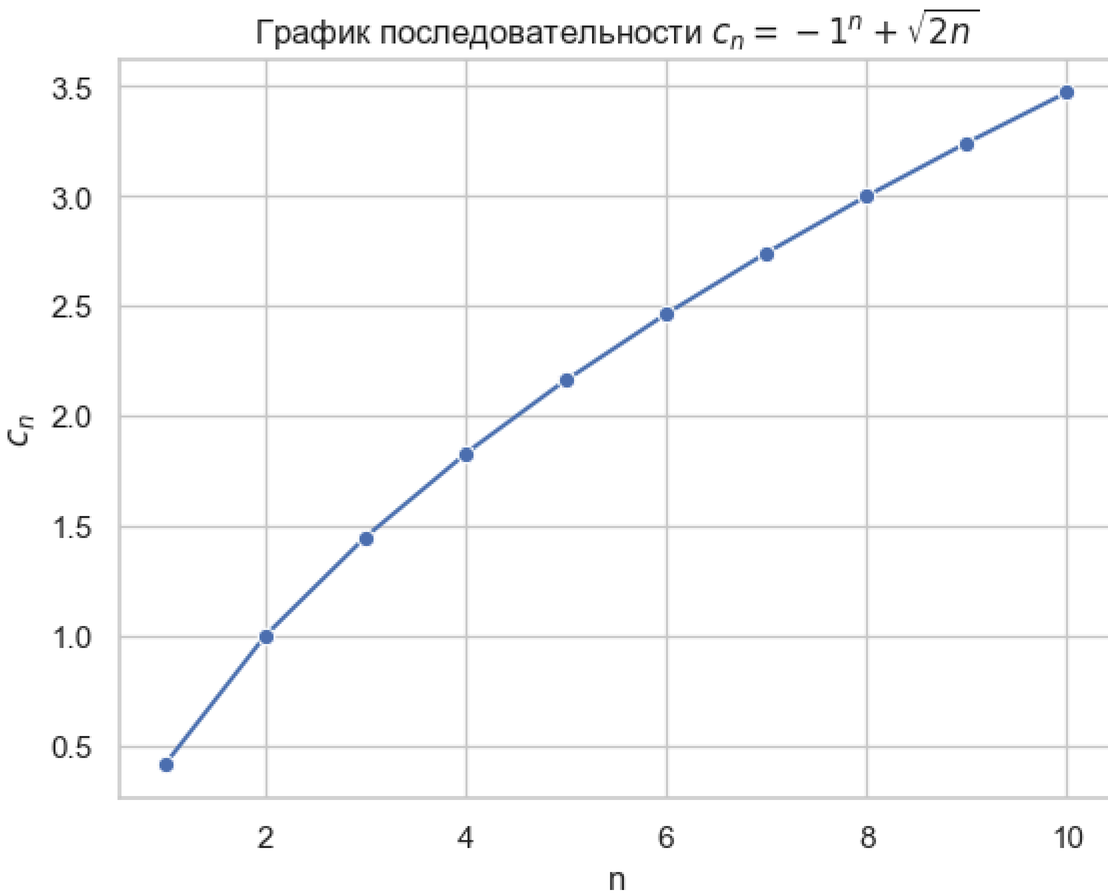
✓ **Check**

$\{c_n\}_{n=1}^{\infty} = -1^n + \sqrt{2n}$

Для исследования монотонности последовательности $\{c_n\}$ необходимо сравнить два соседних члена c_n и c_{n+1}:

$c_n = -1^n + \sqrt{2n}$; $c_{n+1} = -1^{n+1} + \sqrt{2n+1}$; $c_{n+1} - c_n$: $c_{n+1} - c_n = \sqrt{2}\left(\sqrt{n+1} - \sqrt{n}\right)$

Следовательно, $c_{n+1} - c_n = \sqrt{2}\left(\sqrt{n+1} - \sqrt{n}\right)$, $c_{n+1} > c_n$, что означает, что последовательность $\{c_n\}$ монотонно возрастает.



Последовательность ограничена снизу числом $\sqrt{2}-1$.

Нахождение пятого по счету члена

Пятый по счету член последовательности $\{c_n\}$ соответствует $n=5$: $c_5 = \sqrt{10}-1$

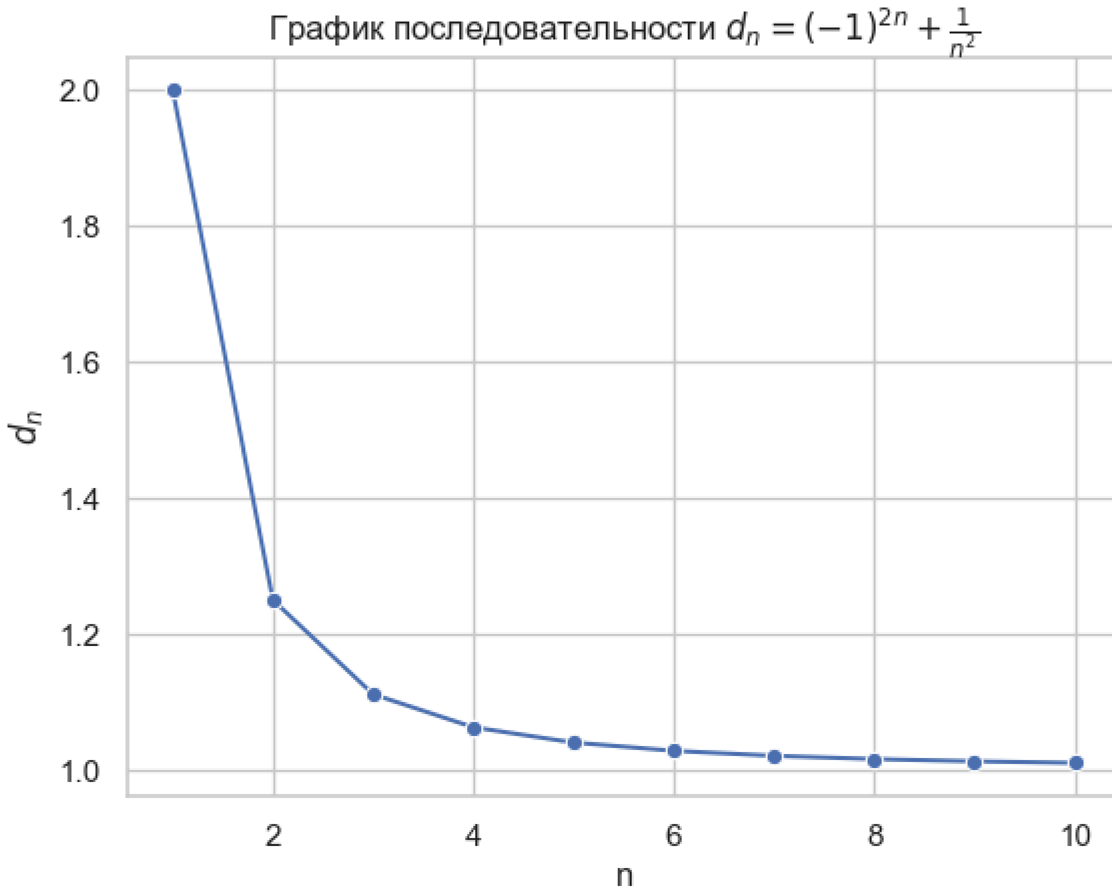
Ответ:

1. Последовательность $\{c_n\}$ монотонно возрастает.
2. Последовательность $\{c_n\}$ ограничена снизу числом $\sqrt{2}-1$.
3. Пятый по счету член последовательности равен $\sqrt{10}-1$.

✓ **Check**

$\{d_n\}_{n=1}^{\infty} = (-1)^{2n} + \frac{1}{n^2}$

Последовательность d_n монотонно убывает.



Последовательность ограничена сверху числом 2, сверху 1

Нахождение пятого по счету члена

Пятый по счету член последовательности $\{d_n\}$ соответствует $n = 5$: $d_5 = \frac{26}{25}$

Ответ:

1. Последовательность $\{d_n\}$ монотонно убывает.
2. Последовательность $\{d_n\}$ ограничена сверху числом 2, снизу 1.
3. Пятый по счету член последовательности равен $\frac{26}{25}$.

Question

Найти 12-й член заданной неявно последовательности:

$$a_1 = 128; a_{n+1} - a_n = 6$$

Решение:

$$a_{12} = a_1 + (n-1) \cdot 6 = 128 + (12-1) \cdot 6 = 194$$

Ответ: 12-й член заданной неявно последовательности $a_{n+1} - a_n = 6$ равен **194**.

Question

*На языке Python предложить алгоритм вычисляющий численно предел с точностью $\epsilon = 10^{-7}$

$$\lim_{n\to\infty} \frac{n}{(n!)^{1/n}}$$

```
import math

def calculate_limit(epsilon=1e-6, max_iterations=100):
    n = 1
    prev_a_n = 0

    for i in range(max_iterations):
        # Вычисляем n!
        n_factorial = math.factorial(n)

        # Вычисляем (sqrt(n!))^(1/n)
        sqrt_n_factorial = math.sqrt(n_factorial)
        sqrt_n_factorial_pow_1_n = sqrt_n_factorial ** (1 / n)

        # Вычисляем a_n
        a_n = n / sqrt_n_factorial_pow_1_n

        # Проверяем условие сходимости
        if abs(a_n - prev_a_n) < epsilon:
            print(f"Предел найден: a_n = {a_n} при n = {n}")
            return a_n

        # Обновляем предыдущее значение a_n
        prev_a_n = a_n

        # Увеличиваем n
        n += 1

    print(f"Предел не найден за {max_iterations} итераций. Последнее значение
a_n = {a_n} при n = {n}")
    return a_n
```

```
# Вызываем функцию для вычисления предела
calculate_limit()
```

⚠ Attention

Предел не найден за 100 итераций. Последнее значение $a_n = 16.22370279204587$ при $n = 101$

При $n = 1000$ программа возвращает ошибку `<OverflowError: int too large to convert to float>`

Давайте использовать приближенную формулу Стирлинга для факториала:

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

Чтобы избежать переполнения при больших n, воспользуемся логарифмом этой формулы. Применяя логарифм к обеим сторонам, получаем:

$$\ln(n!) \approx \ln\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right)$$

Используем свойства логарифмов:

1. Логарифм произведения равен сумме логарифмов: $\ln(ab) = \ln a + \ln b$.
2. Логарифм степени: $\ln\left(a^b\right) = b \ln a$.

Тогда: $\ln(n!) \approx \ln\left(\sqrt{2\pi n}\right) + \ln\left(\left(\frac{n}{e}\right)^n\right)$

Рассмотрим каждое слагаемое отдельно.

Для первого слагаемого: $\ln\left(\sqrt{2\pi n}\right) = \frac{1}{2}\ln(2\pi n)$

Для второго слагаемого: $\ln\left(\left(\frac{n}{e}\right)^n\right) = n \ln\left(\frac{n}{e}\right)$

Используем свойство логарифмов: $\ln\left(\frac{n}{e}\right) = \ln(n) - \ln(e)$. Поскольку $\ln(e) = 1$, получаем: $n \ln\left(\frac{n}{e}\right) = n(\ln(n) - 1)$

Теперь подставим все обратно: $\ln(n!) \approx \frac{1}{2}\ln(2\pi n) + n(\ln(n) - 1)$

Полученное выражение для $\ln(n!)$ в коде выглядит как:

```
log_stirling_factorial = 0.5 * math.log(2 * math.pi * n) + n * (math.log(n) -
1)
```

Это приближение позволяет вычислять факториал для больших n без переполнения, работая с логарифмами.

```
def calculate_limit_stirling(epsilon=1e-6, max_iterations=5000):
    n = 1
    prev_a_n = 0

    for i in range(max_iterations):
        # Вычисляем log(n!)
        log_stirling_factorial = 0.5 * math.log(2 * math.pi * n) + n *
(math.log(n) - 1)

        # Переход к выражению n / (n!)^(1/n) в логарифмической форме
        a_n = math.exp(math.log(n) - log_stirling_factorial / n)

        # Проверка условия сходимости
        if abs(a_n - prev_a_n) < epsilon:
            print(f"Предел найден: a_n = {a_n} при n = {n}")
            return a_n

        # Обновляем предыдущее значение a_n
        prev_a_n = a_n

        # Увеличиваем n
        n += 1

    print(
        f"Предел не найден за {max_iterations} итераций. Последнее значение
a_n = {a_n} при n = {n}"
    )
    return a_n


# Вызываем функцию для вычисления предела
calculate_limit_stirling()
```

✓ **Success**

Предел найден: $a_n = 2.7143969719151477$ при $n = 3495$

Question

*Предложить оптимизацию алгоритма, полученного в задании 3, ускоряющую его сходимость.

Для ускорения сходимости последовательностей к их пределу существуют такие методы как **метод Ричардсона** или **метод Айткена**. Их основная цель — добиться нужной точности при меньшем количестве итераций. Это достигается за счёт исключения или сглаживания медленно сходящихся членов последовательности.

```
import math

def stirling_log_factorial(n):
    """Вычисляет логарифм факториала по формуле Стирлинга."""
    if n == 0:
        return 0  # Логарифм факториала от 0 равен 0
    return 0.5 * math.log(2 * math.pi) + (n + 0.5) * math.log(n) - n

def a_n(n):
    """Последовательность, сходящаяся к e."""
    log_factorial = stirling_log_factorial(n)
    return n / math.exp(log_factorial / n)
```

```
def richardson_limit(epsilon=1e-2, max_iterations=10000):
    """Вычисляет предел с использованием метода Ричардсона."""
    n = 1
    prev_values = []
    iterations = 0

    while iterations < max_iterations:
        current_value = a_n(n)

        prev_values.append(current_value)

        if len(prev_values) >= 2:
            # Применяем метод Ричардсона
            value1 = prev_values[-2]
            value2 = prev_values[-1]
            richardson_value = value2 + (value2 - value1) / ((n / (n - 1)) -
1)

            if abs(richardson_value - prev_values[-1]) < epsilon:
                return richardson_value, iterations + 1
```

```python
        n += 1
        iterations += 1

    return current_value, iterations
```

✓ Success

Результат: 2.716975958739938, Количество итераций: 1056

```python
def aitken_extrapolation(epsilon=1e-2, max_iterations=1000):
    """Метод Айткена для экстраполяции последовательности a_n к пределу."""
    n = 1
    a_n_prev = a_n(n)
    a_n_curr = a_n(n + 1)
    iterations = 0

    while iterations < max_iterations:
        a_n_next = a_n(n + 2)

        # Айткеновская экстраполяция
        aitken_value = a_n_curr - (a_n_curr - a_n_prev) ** 2 / (a_n_next - 2 *
a_n_curr + a_n_prev)

        # Проверяем условие сходимости
        if abs(aitken_value - a_n_curr) < epsilon:
            return aitken_value, iterations + 1

        # Обновляем значения для следующей итерации
        a_n_prev, a_n_curr = a_n_curr, a_n_next
        n += 1
        iterations += 1

    return aitken_value, iterations
```

✓ Success

Результат: 2.7071422828586558, Количество итераций: 517

Для удобства объединим полученные результаты в таблицу.

Метод	**Точность** ϵ	**Количество итераций**	**Полученное значение**
Метод последовательных приближений	10^{-6}	3495	2.714
Метод Айткена (Δ^2-экстраполяция)	10^{-2}	1056	2.7071
Метод Ричардсона	10^{-2}	517	2.717