

Решение домашнего задания к уроку “Множество. Последовательность”

Тема “Введение в математических анализ”

1. Задание

Как относятся друг к другу множество и последовательность? (в ответе использовать слова типа: часть, целое, общее, частное, родитель, дочерний субъект и т.д.)

Ответ:
 Последовательность - набор элементов множества, в котором каждому натуральному числу соответствует элемент \Rightarrow Последовательность - это дочерний субъект множества

2. Задание

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

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

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

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

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

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

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

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

1. $\forall y \in [0; 1] : \operatorname{sgn}(y) = 1$
 Для любого y из отрезка от нуля до единицы, значение функции $\operatorname{sgn}()$ равно 1.
Отрицание: $\exists y \in [0; 1] : \operatorname{sgn}(y) \neq 1$
Истинность исходного выражения: неверно для граничной точки 0

1. $\forall n \in \mathbb{N} > 2 : \exists x, y, z \in \mathbb{N} : x^n = y^n + z^n$
 Для любого натурального n больше двух, существуют такие три натуральных числа x, y и z , что выполнено равенство $x^n = y^n + z^n$.
Отрицание: $\exists n \in \mathbb{N} > 2 : \forall x, y, z \in \mathbb{N} : x^n \neq y^n + z^n$
Истинность исходного выражения: неверно (теорема Ферма)

1. $\forall x \in \mathbb{R} \exists X \in \mathbb{R} : X > x$
 Для любого действительного числа найдется такое подмножество множества действительных чисел, каждый элемент которого будет больше x .
Отрицание: $\exists x \in \mathbb{R} \forall X \in \mathbb{R} : X \leq x$
Истинность исходного выражения: верно

1. $\forall x \in \mathbb{C} \nexists y \in \mathbb{C} : x > y || x < y$
 Для всех комплексных чисел x не существует комплексных чисел y , таких, что x больше y или x меньше y .
Отрицание: $\exists x \in \mathbb{C} \exists y \in \mathbb{C} : x \leq y || x \geq y$
Истинность исходного выражения: верно (комплексные числа не подлежат сравнению)

1. $\forall y \in [0; \frac{\pi}{2}] \exists \varepsilon > 0 : \sin y < \sin(y + \varepsilon)$
Для всех y из отрезка $[0; \frac{\pi}{2}]$ существует ε больше нуля, такой, что выполнено неравенство $\sin y < \sin(y + \varepsilon)$.
Отрицание: $\exists y \in [0; \frac{\pi}{2}] \forall \varepsilon \leq 0 : \sin y \geq \sin(y + \varepsilon)$
Истинность исходного выражения: неверно для граничной точки $\frac{\pi}{2}$

1. $\forall y \in [0; \pi) \exists \varepsilon > 0 : \cos y > \cos(y + \varepsilon)$
Для всех y из полуинтервала $[0; \pi)$ существует ε больше нуля, такой, что выполнено неравенство $\cos y > \cos(y + \varepsilon)$.
Отрицание: $\exists y \in [0; \pi) \forall \varepsilon \leq 0 : \cos y \leq \cos(y + \varepsilon)$
Истинность исходного выражения: верно

1. $\exists x : x \notin \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}\}$
Существует число x , не принадлежащее к множествам натуральных, целых, рациональных, вещественных и комплексных чисел.
Отрицание: $\forall x : x \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}\}$
Истинность исходного выражения: верно (гиперкомплексные числа $x \in \{\mathbb{H}, \mathbb{O}, \mathbb{S}\}$)

Тема “Множество”

1. Задание

Даны три множества a, b и c . Необходимо выполнить все изученные виды бинарных операций над всеми комбинациями множеств.

Пусть:
 $a = \{1, 2, 3\}$
 $b = \{3, 4, 5, 6\}$
 $c = \{-1, 0, 1\}$
Пересечение:
 $a \cap b = \{1, 2, 3\} \cap \{3, 4, 5, 6\} = \{3\}$
 $a \cap c = \{1, 2, 3\} \cap \{-1, 0, 1\} = \{1\}$
 $b \cap c = \{3, 4, 5, 6\} \cap \{-1, 0, 1\} = \emptyset$
 $a \cap b \cap c = \{1, 2, 3\} \cap \{3, 4, 5, 6\} \cap \{-1, 0, 1\} = \emptyset$
Объединение:
 $a \cup b = \{1, 2, 3\} \cup \{3, 4, 5, 6\} = \{1, 2, 3, 4, 5, 6\}$
 $a \cup c = \{1, 2, 3\} \cup \{-1, 0, 1\} = \{-1, 0, 1, 2, 3\}$
 $b \cup c = \{3, 4, 5, 6\} \cup \{-1, 0, 1\} = \{-1, 0, 1, 3, 4, 5, 6\}$
 $a \cup b \cup c = \{1, 2, 3\} \cup \{3, 4, 5, 6\} \cup \{-1, 0, 1\} = \{-1, 0, 1, 2, 3, 4, 5, 6\}$
Разность:
 $a \setminus b = \{1, 2, 3\} \setminus \{3, 4, 5, 6\} = \{1, 2\}$
 $b \setminus a = \{3, 4, 5, 6\} \setminus \{1, 2, 3\} = \{4, 5, 6\}$
 $a \setminus c = \{1, 2, 3\} \setminus \{-1, 0, 1\} = \{2, 3\}$
 $c \setminus a = \{-1, 0, 1\} \setminus \{1, 2, 3\} = \{-1, 0\}$
 $b \setminus c = \{3, 4, 5, 6\} \setminus \{-1, 0, 1\} = \{3, 4, 5, 6\}$
 $c \setminus b = \{-1, 0, 1\} \setminus \{3, 4, 5, 6\} = \{-1, 0, 1\}$
Симметрическая разность:
 $a \triangle b = \{1, 2, 3\} \triangle \{3, 4, 5, 6\} = \{1, 2, 4, 5, 6\}$
 $a \triangle c = \{1, 2, 3\} \triangle \{-1, 0, 1\} = \{-1, 0, 2, 3\}$
 $b \triangle c = \{3, 4, 5, 6\} \triangle \{-1, 0, 1\} = \{-1, 0, 1, 3, 4, 5, 6\}$
 $a \triangle b \triangle c = \{1, 2, 3\} \triangle \{3, 4, 5, 6\} \triangle \{-1, 0, 1\} = \{-1, 0, 2, 4, 5, 6\}$
Декартово произведение:
 $a \times b = \{1, 2, 3\} \times \{3, 4, 5, 6\} = \{(1, 3), (2, 3), (3, 3), (1, 4), (2, 4), (3, 4), (1, 5), (2, 5), (3, 5), (1, 6), (2, 6), (3, 6)\}$
 $a \times c = \{1, 2, 3\} \times \{-1, 0, 1\} = \{(1, -1), (2, -1), (3, -1), (1, 0), (2, 0), (3, 0), (1, 1), (2, 1), (3, 1)\}$
 $b \times c = \{3, 4, 5, 6\} \times \{-1, 0, 1\} = \{(3, -1), (4, -1), (5, -1), (6, -1), (3, 0), (4, 0), (5, 0), (6, 0), (3, 1), (4, 1), (5, 1), (6, 1)\}$

2. Задание

*Выполнить задание 1 на языке Python

```
In [1]: import itertools
a = {1, 2, 3}
b = {3, 4, 5, 6}
c = {-1, 0, 1}
print(f'a = {a}, b = {b}, c = {c}')
print()
print(f'Пересечение a и b: {a.intersection(b)}')
print(f'Пересечение a и c: {a.intersection(c)}')
print(f'Пересечение b и c: {b.intersection(c)}')
print(f'Пересечение a, b и c: {a.intersection(b).intersection(c)}')
print()
print(f'Объединение a и b: {a.union(b)}')
print(f'Объединение a и c: {a.union(c)}')
print(f'Объединение b и c: {b.union(c)}')
print(f'Объединение a, b и c: {a.union(b).union(c)}')
print()
print(f'Разность a и b: {a.difference(b)}')
print(f'Разность b и a: {b.difference(a)}')
print(f'Разность a и c: {a.difference(c)}')
print(f'Разность c и a: {c.difference(a)}')
print(f'Разность b и c: {b.difference(c)}')
print(f'Разность c и b: {c.difference(b)}')
print()
print(f'Симметрическая разность a и b: {a.symmetric_difference(b)}')
print(f'Симметрическая разность a и c: {a.symmetric_difference(c)}')
print(f'Симметрическая разность b и c: {b.symmetric_difference(c)}')
print(f'Симметрическая разность a, b и c: {a.symmetric_difference(b).symmetric_difference(c)}')
print()
print(f'Декартово произведение a и b: {set(itertools.product(*[a,b]))}')
print(f'Декартово произведение a и c: {set(itertools.product(*[a,c]))}')
print(f'Декартово произведение b и c: {set(itertools.product(*[b,c]))}')
```

a = {1, 2, 3}, b = {3, 4, 5, 6}, c = {0, 1, -1}

Пересечение a и b: {3}
Пересечение a и c: {1}
Пересечение b и c: set()
Пересечение a, b и c: set()

Объединение a и b: {1, 2, 3, 4, 5, 6}
Объединение a и c: {0, 1, 2, 3, -1}
Объединение b и c: {0, 1, 3, 4, 5, 6, -1}
Объединение a, b и c: {0, 1, 2, 3, 4, 5, 6, -1}

Разность a и b: {1, 2}
Разность b и a: {4, 5, 6}
Разность a и c: {2, 3}
Разность c и a: {0, -1}
Разность b и c: {3, 4, 5, 6}
Разность c и b: {0, 1, -1}

Симметрическая разность a и b: {1, 2, 4, 5, 6}
Симметрическая разность a и c: {0, 2, 3, -1}
Симметрическая разность b и c: {0, 1, 3, 4, 5, 6, -1}
Симметрическая разность a, b и c: {0, 2, 4, 5, 6, -1}

Декартово произведение a и b: {(1, 3), (2, 6), (3, 3), (1, 4), (1, 5), (1, 6), (2, 3), (3, 6), (2, 5), (3, 4), (2, 4), (3, 5)}
Декартово произведение a и c: {(3, 0), (2, -1), (3, 1), (2, 1), (3, -1), (2, 0), (1, 0), (1, -1), (1, 1)}
Декартово произведение b и c: {(5, -1), (3, 0), (6, 1), (3, 1), (6, 0), (3, -1), (6, -1), (5, 0), (5, 1), (4, 1), (4, -1), (4, 0)}

Тема “Последовательность”

1. Задание

Даны 4 последовательности. Необходимо:

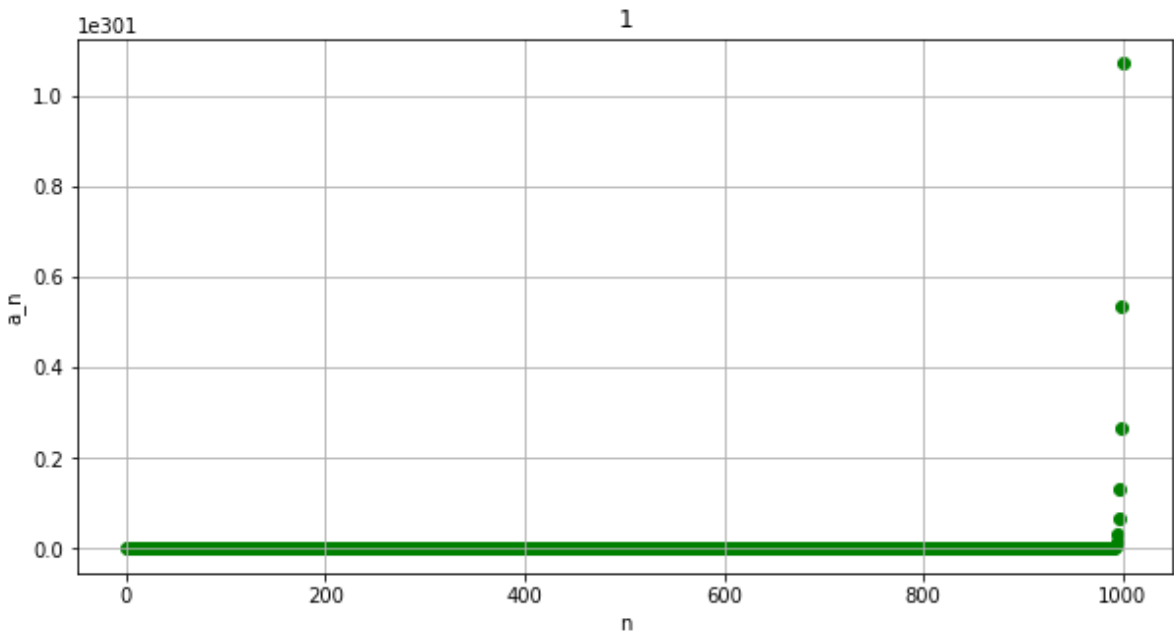
- исследовать их на монотонность;
- исследовать на ограниченность;
- найти пятый по счету член.

```
In [2]: import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline

def plot_seq(n, a, title, color):
    plt.figure(figsize=(10,5))
    plt.scatter(n, a, c=color)
    plt.title(title)
    plt.xlabel('n')
    plt.ylabel('a_n')
    plt.grid()
    plt.show()
```

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

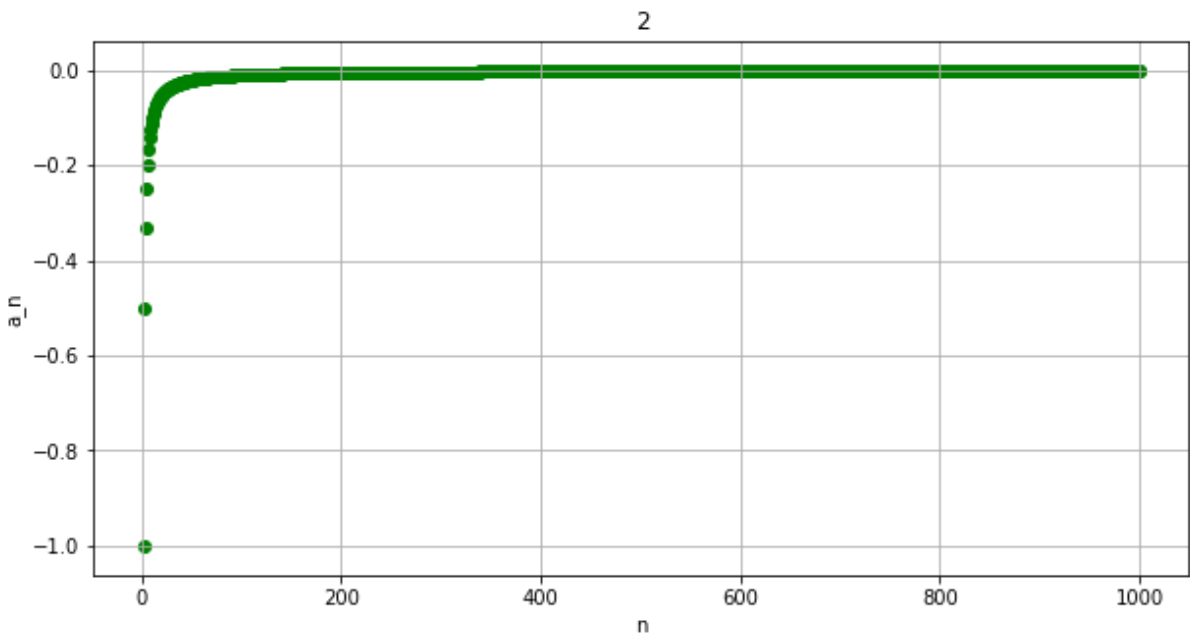
```
In [3]: n = np.linspace(1, 1000, 1000)
plot_seq(n,2**n-n, 'r', 'g')
```



- а. Монотонность:** $a_{n+1} - a_n = 2^{n+1} - (n + 1) - (2^n - n) = 2^{n+1} - 2^n - n - 1 + n = 2^n(2 - 1) - 1 = 2^n - 1 > 0$ при $n \in [1, \infty) \Rightarrow$ последовательность возрастающая.
- б. Ограниченность:** $a_1 = 1$, т.к. последовательность возрастающая, то она ограничена снизу 1. Т.к. 2^n растет сильно быстрее, чем n , то исходная последовательность не ограничена сверху.
- с. Пятый по счету член:** $a_5 = 2^5 - 5 = 32 - 5 = 27$.

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

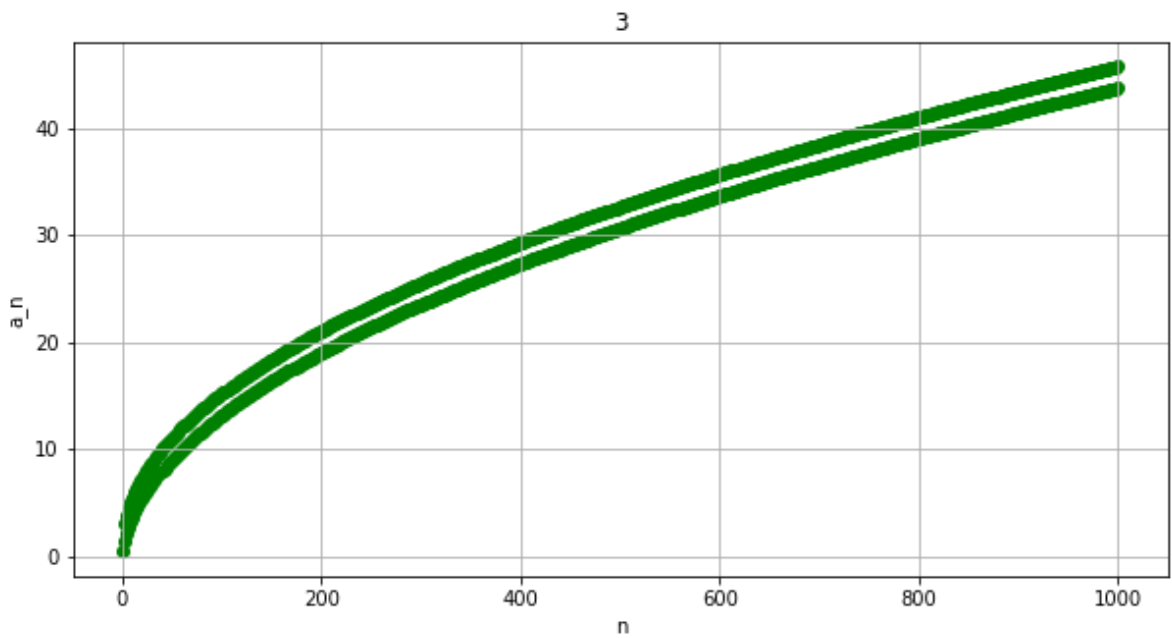
```
In [4]: n = np.linspace(2, 1000, 999)
plot_seq(n,1/(1-n), 'r', 'g')
```



- а. Монотонность:** $b_{n+1} - b_n = \frac{1}{1-(n+1)} - \frac{1}{1-n} = \frac{1}{-n} - \frac{1}{1-n} = \frac{1}{n-1} - \frac{1}{n} = \frac{n-(n-1)}{n(n-1)} = \frac{1}{n^2-n} = \frac{1}{n} \cdot \frac{1}{n-1} > 0$ при $n \in [2, \infty) \Rightarrow$ последовательность возрастающая.
- б. Ограниченность:** $b_2 = -1$, т.к. последовательность возрастающая, то она ограничена снизу -1. Также последовательность ограничена 0 сверху.
- с. Пятый по счету член:** $b_5 = \frac{1}{1-6} = \frac{1}{-5} = -0.2$ (т.к. n начинается с 2, а не с 1).

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

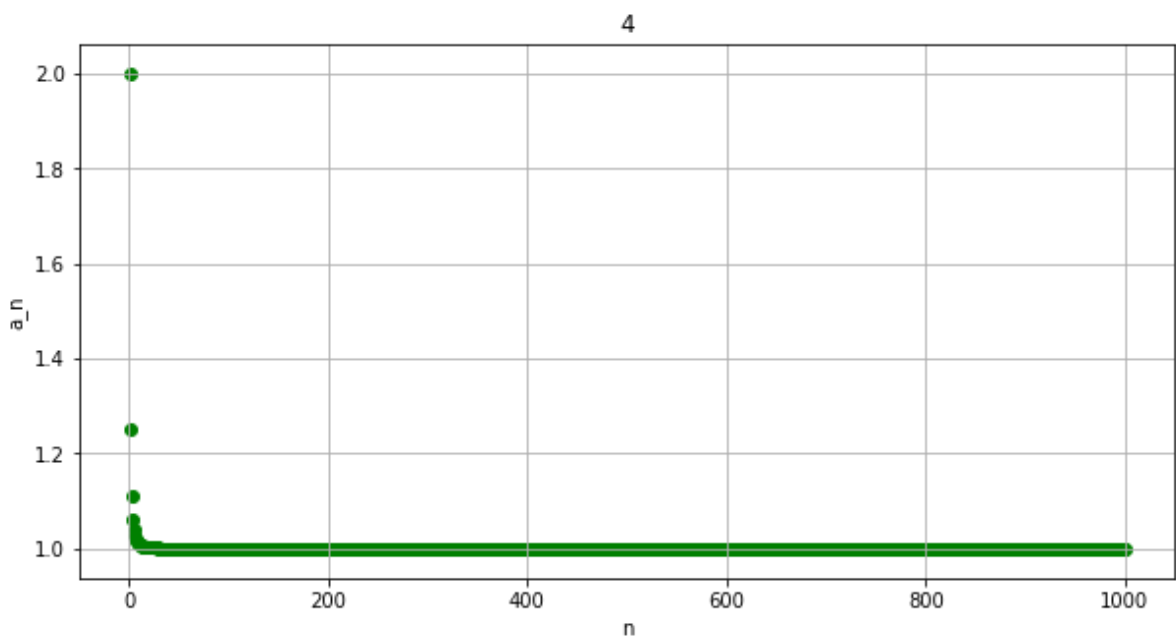
```
In [5]: n = np.linspace(1, 1000, 1000)
plot_seq(n, (-1)**n+(2*n)**(0.5), '3', 'g')
```



- a. Монотонность:** $c_{n+1} - c_n = -1^{n+1} + \sqrt{2(n+1)} - (-1^n + \sqrt{2n}) = -1^n(-1) - (-1^n) + \sqrt{2(n+1)} - \sqrt{2n} = -2(-1^n) + \sqrt{2(n+1)} - \sqrt{2n} < 0$ при $n \in [1, \infty)$ и четных n и > 0 при нечетных $n \Rightarrow$ последовательность не является ни возрастающей, ни убывающей (я бы назвала ее "осциллирующей", если так можно сказать).
- b. Ограниченность:** $c_1 = -1 + \sqrt{2}$, $c_2 = 3$, $c_3 = -1 + \sqrt{6}$, $c_4 = 2\sqrt{2} + 1$ Несмотря на то, что значения членов последовательность "прыгают" то вверх, то вниз, $c_{n+2} > c_n$ и $c_{n+3} > c_{n+1}$, т.е. последовательность можно как бы разбить на две возрастающие подпоследовательности(видно на графике) \Rightarrow глобально значения членов последовательности увеличиваются \Rightarrow исходная последовательность ограничена снизу $\sqrt{2} - 1$ и не ограничена сверху.
- с. Пятый по счету член:** $c_5 = -1^5 + \sqrt{2 * 5} = \sqrt{10} - 1$.

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

```
In [6]: n = np.linspace(1, 1000, 1000)
plot_seq(n, (-1)**(2*n)+1/(n*n), '4', 'g')
```



- a. Монотонность:** $d_{n+1} - d_n = -1^{2(n+1)} + \frac{1}{(n+1)^2} - (-1^{2n} + \frac{1}{n^2}) = \frac{1}{(n+1)^2} - \frac{1}{n^2} = \frac{n^2-(n+1)^2}{n^2(n+1)^2} = \frac{n^2-n^2-2n-1}{n^2(n+1)^2} = -\frac{2n+1}{n^2(n+1)^2} < 0$ при $n \in [1, \infty) \Rightarrow$ последовательность убывающая.
- b. Ограниченность:** $d_1 = 2$, т.к. последовательность убывающая, то она ограничена сверху 2. Т.к. $d_n = -1^{2n} + \frac{1}{n^2} = \frac{n^2+1}{n^2} > 0 \forall n$, то последовательность ограничена снизу 0.
- с. Пятый по счету член:** $d_5 = -1^{2*5} + \frac{1}{5^2} = 1 + \frac{1}{25} = \frac{26}{25} = 1.04$.

2. Задание

Найти 12-й член заданной неявно последовательности

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

Решение: $a_{n+1} = a_n + 6$

```
In [7]: d = {}
d[1] = 128
for i in range(1,12):
    i += 1
    d[i] = d[i - 1] + 6
print(d)
print(f'12-й член последовательности равен {d[12]}')
```

{1: 128, 2: 134, 3: 140, 4: 146, 5: 152, 6: 158, 7: 164, 8: 170, 9: 176, 10: 182, 11: 188, 12: 194}
12-й член последовательности равен 194

Ответ: 194.

3. Задание

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

$$\lim_{n \rightarrow +\infty} \frac{n}{\sqrt[n]{n!}}$$

```
In [8]: # шаг n += 1000 для ускорения вычисления

FACTORIAL_DICT = {0:1, 1:1} # глобальная переменная со значениями факториалов
from decimal import Decimal

# Функция вычисляет факториал n
# В случае, когда в глобальной переменной FACTORIAL_DICT нет требуемого значения, смотрит предыдущее значение с шагом 1000
# и записывает в FACTORIAL_DICT два следующих соседних значения
def get_factorial(n):
    if n in FACTORIAL_DICT.keys():
        return FACTORIAL_DICT[n]
    else:
        res = get_factorial(n - 1000)
        for i in range(1000):
            res = res * (n - i)
        FACTORIAL_DICT[n] = res
        FACTORIAL_DICT[n+1] = res * (n + 1)
        return res

# Функция возвращает значение последовательности для переданного n
def get_func(n):
    ch = n
    zn = Decimal(get_factorial(n))**Decimal(1 / n)
    return ch / zn

eps = 10**(-7) # требуемая точность
k = 1 # текущая точность
n = 0 # текущий элемент последовательности
a = 1 # текущее значение предела
while True:
    n = n + 1000
    a_prev = get_func(n)
    a_next = get_func(n+1)
    k = a_next - a_prev
    a = a_prev
    if n%1000==0:
        print(n, k, a)
    if k < eps:
        break
print()
print(f'Значение предела: {a}, точность {k}, достигнута при n={n}')
```

1000 0.000010472146180777131958378 2.706421007184342557277141577
2000 0.000002859327469086192613444 2.711875018209429156052116326
3000 0.000001333027694301686930710 2.713825554813455272683974315
4000 7.74565761952716316300E-7 2.714841314428223173110228344
5000 5.07975472250971495424E-7 2.715468476345699207309010195
6000 3.59703133877890200831E-7 2.715895903158288476553578591
7000 2.68579564757467224776E-7 2.716206714467579548701815152
8000 2.08485977410770020252E-7 2.716443346816620442398647810
9000 1.66717979736027547907E-7 2.716629785153760260466919320
10000 1.36481499695707106743E-7 2.716780632436130643035174488
11000 1.13870679793921831847E-7 2.716905300301416414503683182
12000 9.6508161008163630043E-8 2.717010134027562283324509418

Значение предела: 2.717010134027562283324509418, точность 9.6508161008163630043E-8, достигнута при n=12000

4. Задание

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

В качестве оптимизации используем формулу Стирлинга:

lim_{n -> +inf} n! / (sqrt(2*pi*n) * (n/e)^n) = 1

То есть, при n -> +inf можно заменить значение факториала на n! ~ sqrt(2*pi*n) * (n/e)^n

```
In [9]: from decimal import Decimal
import math

# Здесь используем формулу Стирлинга
def get_factorial(n):
    return ((Decimal(n / math.e))**Decimal(n)) * Decimal((math.pi * 2 * n)**0.5)

def get_func(n):
    ch = n
    zn = Decimal(get_factorial(n))**Decimal(1 / n)
    return ch / zn

eps = 10**(-7)
k = 1
n = 0
a = 1
while True:
    n = n + 500
    a_prev = get_func(n)
    a_next = get_func(n+1)
    k = a_next - a_prev
    a = a_next
    if n%500==0:
        print(n, k, a)
    if k < eps:
        break
print()
print(f'Значение предела: {a}, точность {k}, достигнута при n={n}')
```

500 0.000037963508197388786702666 2.696518761907988072408942547
1000 0.000010471696657160350161904 2.706431704416085307316210170
1500 0.000004905711546918484590577 2.710007613790310360533554111
2000 0.000002859271073503598263545 2.711877933977898767504266886
2500 0.000001879299767332423404369 2.713036013687631967121974644
3000 0.000001333010962971919768341 2.713826912952432301723213846
3500 9.96680708741664745924E-7 2.714402960073219010607005563
4000 7.74558698715491793147E-7 2.714842103126720185883070916
4500 6.19985783689028383095E-7 2.715188477269692539601588747
5000 5.07971854388365535887E-7 2.715468993369115047593981340
5500 4.24132019388461404719E-7 2.715701017167209789747360643
6000 3.59701039901461808637E-7 2.715896269146124088857867670
6500 3.09085761828472949112E-7 2.716062953183203874115221804
7000 2.68578245417025036777E-7 2.716206987665224097745531059
7500 2.35640067233066606704E-7 2.716332752187385536816042017
8000 2.08485093788659958460E-7 2.716443558838749616416797691
8500 1.85826399370367155904E-7 2.716541959898248660561303245
9000 1.66717358729462685683E-7 2.716629954666005610684103878
9500 1.50448859337173298236E-7 2.716709132065975869959065391
10000 1.36481047240276158713E-7 2.716780771181161604843020985
10500 1.24396964231695550267E-7 2.716845913544343153021041199
11000 1.13870339855373569218E-7 2.716905416042902909934111464
11500 1.04642885829757598208E-7 2.716959990253507182987608691
12000 9.6507899452450936358E-8 2.717010232107805200483324524

Значение предела: 2.717010232107805200483324524, точность 9.6507899452450936358E-8, достигнута при n=12000

```
In [ ]:
```