

. Текстовый разбор домашки 3

DZ_253_1

Задача № 1 (18192)

Напишите программу, которая перебирает целые числа, бóльшие 1 000 000, в порядке возрастания и ищет среди них те, которые имеют ровно 3 простых делителя. В ответе запишите 5 наименьших таких чисел в порядке возрастания. Справа от каждого такого числа укажите его наибольший простой делитель.

Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h0m0s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Затем перебираем все числа от 1 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d как делитель i , так и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 1000001 до 1000100. Для каждого числа из диапазона находим его делители с помощью функции $\text{div}(x)$. Далее проверяем, какие из делителей являются простыми, то есть делятся только на единицу и на себя (длина множества делителей числа i равна 2). Отбираем те числа, у которых ровно 3 таких делителя, то есть длина множества d делителей числа x равно 3 и выводим на экран само число и его наибольший простой делитель (это последний элемент списка d , то есть $d[-1]$). Скопируем в ответ первые пять пар чисел.

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(1_000_001, 1_000_100):
    d = [i for i in div(x) if len(div(i))==2]
    if len(d)==3:
        print(x, d[-1])
```

Ответ:

1000002	166667
1000004	89
1000006	71429
1000012	19231
1000013	383

Задача №2 (19775)

Обозначим через S сумму простых делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то считаем значение S равным нулю. Напишите программу, которая перебирает целые числа, большие 32 500 000 в порядке возрастания и ищет среди них такие, для которых значение S не равно нулю и кратно 145.

Программа должна найти первые 7 таких чисел. Для каждого из них в отдельной строке сначала выводится само число, затем значение S . Строки выводятся в порядке возрастания найденных чисел. Количество строк для записи ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h2m55s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Поскольку нет необходимости считать 1 и само число (нам нужны только нетривиальные делители), будем перебирать все числа от 2 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d делитель i и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 32500000 до 32501000. Для каждого числа из диапазона находим его простые делители с помощью функции $\text{div}(x)$, причем в множество делителей d попадают только простые делители (длина множества делителей числа i равна 0). Считаем сумму этих простых делителей в переменной S . Если сумма S делителей не равна нулю и делится на 145 (то есть $S \% 145 == 0$), то выводим на экран число x и сумму его простых делителей

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(32_500_001, 32_501_000):
    d = [i for i in div(x) if len(div(i))==0]
    S = sum(d)
    if S!=0 and S%145==0:
        print(x, S)
```

Ответ:

32500280	2755
32500301	58290
32500440	1450
32500623	17545
32500665	722245
32500700	7975
32500834	4785

Задача № 3 (1393)

Обозначим через F целую часть среднего арифметического всех простых делителей целого числа, не считая самого числа. Если таких делителей у числа нет, то считаем значение F равным нулю.

Напишите программу, которая перебирает целые числа, большие 650000, в порядке возрастания и ищет среди них такие, для которых значение F при делении на 37 даёт в остатке 23.

Выведите первые 4 найденных числа в порядке возрастания и справа от каждого числа – соответствующее значение F .

Количество строк для записи ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h5m00s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Поскольку нет необходимости считать 1 и само число (нам нужны только нетривиальные делители), будем перебирать все числа от 2 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d делитель i и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 650001 до 651000. Для каждого числа из диапазона находим его простые делители с помощью функции $\text{div}(x)$, причем в множество делителей d попадают только простые делители (длина множества делителей числа i равна 0). Если простые делители числа x найдены (то есть список d не пуст), то рассчитывается их среднее арифметическое F как частное от суммы простых делителей и их количества. Если среднее арифметическое F делится на 37 с остатком 23 (то есть $F \% 37 == 23$), то выводим на экран число x и соответствующее ему среднее арифметическое F . Первые 4 пары чисел копируем в ответ.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(650_001, 651_000):
    d = [i for i in div(x) if len(div(i))==0]
    if len(d)>0:
        F = sum(d)//len(d)
        if F%37==23:
            print(x, F)
```

Ответ:

650090	60
650153	282
650155	3945
650208	134

Telegram: @fast_ege

DZ_253_4

Задача № 4 (19779)

Найдите 5 чисел больших 55 000 000, таких, что среди их простых делителей есть число, оканчивающееся на 777, при не равное самому числу.

В качестве ответа приведите 4 наименьших числа, соответствующих условию.

Формат вывода: для каждого из 4 таких найденных чисел в отдельной строке сначала выводится само число, затем минимальный простой делитель, оканчивающийся на 777, не равный самому числу.

Количество строк для записи ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h7m10s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Поскольку нет необходимости считать 1 и само число (нам нужны только нетривиальные делители), будем перебирать все числа от 2 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d делитель i и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 55000001 до 55010000. Для каждого числа из диапазона находим его простые делители с помощью функции $\text{div}(x)$, причем в множество делителей d попадают только те делители i , которые являются простыми (длина множества делителей числа i равна 0), которые оканчиваются на 777 ($i \% 1000 == 777$). Если такие делители числа x найдены (то есть список d не пуст), то выводим на экран число x и соответствующий ему минимальный простой делитель, который оканчивается на 777. Первые 4 пары чисел копируем в ответ.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(55_000_001, 55_010_000):
    d = [i for i in div(x) if len(div(i))==0 and i%1000==777]
    if len(d)>0:
        print(x, d[0])
```

Ответ:

55000662	9166777
55001262	2777
55001554	27500777
55001704	1777

Telegram: @fast_ege

DZ_253_5

Задача №5 (2591)

Среди целых чисел, принадлежащих числовому отрезку [125697; 125721], найдите числа, которые представляют собой произведение двух различных простых делителей. Для каждого найденного числа запишите эти два делителя в таблицу на экране с новой строки в порядке возрастания произведения этих двух делителей. Делители в строке таблицы также должны следовать в порядке возрастания.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h9m35s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Затем перебираем все числа от 1 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d как делитель i , так и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 125697 до 125721 (диапазон ограничен числом 125722, которое не включается в вычисления в цикле). Для каждого числа из диапазона находим его делители с помощью функции $\text{div}(x)$, проверяем, какие из делителей являются простыми, то есть делятся только на единицу и на себя (длина множества делителей числа i равна 2, то есть число i делится только на единицу и на само себя). После этого, проверяем состоит ли число x из произведения двух различных простых чисел, минимальных делителей из множества d (то есть первых элементов этого множества, $d[0]$ и $d[1]$). Если да, то выводим на экран эти два простых числа.

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(125_697, 125_722):
    d = [i for i in div(x) if len(div(i))==2]
    if len(d)==2 and d[0]*d[1]==x:
        print(d[0],d[1])
```

Ответ:

7	17957
337	373
2	62851
3	41903
7	17959

Telegram: @fast_ege

DZ_253_6

Задание №6 (5477)

Напишите программу, которая перебирает целые числа, большие 600 000, в порядке возрастания и ищет среди них такие, что они делятся на 6 без остатка и их соседи - простые числа. Для первых шести найденных чисел выведите соседей каждого найденного числа в порядке возрастания.

Соседями числа называются числа, отличные от него на 1.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h11m55s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Затем перебираем все числа от 1 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d как делитель i , так и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 600006 до 601000 с шагом 6, чтобы гарантировать, что проверяемые числа делятся на 6. Мы начинаем с 600,006, потому что оно первое число больше 600,000, которое делится на 6. Для каждого числа x , которое мы рассматриваем в цикле, мы проверяем, что его соседи $x-1$ и $x+1$ – простые числа, то есть делятся только на единицу и на себя (длина множества делителей чисел $x-1$ и $x+1$ равна 2). Если такие соседи числа x находятся, то выводим их на экран. Первые 6 пар чисел копируем в ответ.

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(600_006, 601_000, 6):
    if len(div(x-1)) == 2 and len(div(x+1)) == 2:
        print(x-1, x+1)
```

Ответ:

600071	600073
600167	600169
600239	600241
600317	600319
600359	600361
600401	600403

Telegram: @fast_ege

DZ_253_7

Задание № 7(4901)

Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:

- символ «?» означает ровно одну произвольную цифру;
- символ «*» означает любую последовательность цифр произвольной длины; в том числе «*» может задавать и пустую последовательность.

Найдите наименьшие 7 чисел, удовлетворяющих маске ?6*6*?6 и при этом кратных 6, 7 и 8.

Выведите эти числа в порядке возрастания, справа от каждого числа выведите сумму его делителей.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h14m20s

Решение

Нам потребуется модуль `fnmatch`, пишем `from fnmatch import *`. Далее, нам нужна будет функция для поиска делителей, создадим её. Пишем функцию $\text{div}(x)$, которая находит все делители числа x .

Для этого создадим пустое множество d , в которое будем складывать делители. Затем перебираем все числа от 1 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d как делитель i , так и результат деления x/i . Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

Числа, которые мы ищем, должны быть кратны 6, 7 и 8. Это можно выразить через их наименьшее общее кратное (НОК). НОК чисел 6, 7 и 8 равен 168. Следовательно, мы будем проверять числа, кратные 168. То есть, будем перебирать все числа в диапазоне от 0 до 10^6 с шагом 168, чтобы гарантировать, что проверяемые числа делятся на 6, 7 и 8. Для каждого числа x , которое мы рассматриваем в цикле, мы проверяем его соответствие маске `?6*6*?6` с помощью функции `fnmatch`. Если такие числа x находятся, то выводим на экран их и сумму их делителей (`sum (div(x))`). Первые 7 пар чисел копируем в ответ.

```
from fnmatch import *

def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(0, 10**6, 168):
    if fnmatch(str(x), '?6*6*?6'):
        print(x, sum(div(x)))
```

Ответ:

56616	162240
66696	191040
161616	527744
166656	523264
266616	862680
360696	1094400
366576	1083264

Telegram: @fast_ege

DZ_253_8

Задание №8 (14440)

Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:

- символ «`?`» означает ровно одну произвольную цифру;
- символ «`*`» означает любую последовательность цифр произвольной длины; в том числе «`*`» может задавать и пустую последовательность.

Например, маске `123*405` соответствуют числа `123405` и `12300405`.

Среди натуральных чисел, не превышающих 10^7 , найдите все простые числа, соответствующие маске `31*567?`

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им результаты произведения их цифр.

Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h17m05s

Решение

Нам потребуется модуль `itertools`, пишем `from itertools import *`. Далее, нам нужна будет функция для поиска делителей, создадим её. Пишем функцию `div(x)`, которая находит все делители числа x . Для этого создадим пустое множество `d`, в которое будем складывать делители. Затем перебираем все числа от 1 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество `d` как делитель i , так и результат деления $x//i$. Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей. Затем нам потребуется функция для нахождения произведения всех цифр числа, напомним функцию `pr(x)`. Присваиваем `p` начальное значение 1. Будем перебирать символы из строковой записи числа x и умножать на них число `p`, предварительно превратив в числа (сменив формат на `int`). В результате работы функция будет возвращать нам число `p`, которое является произведением цифр числа x . Создадим список `comb`, куда будем добавлять все возможные комбинации цифр, которые могут оказаться на месте `*`. Их количество может быть равно 0 или 1, потому что наше число ограничено 8 символами (не превышает 10^7).

Далее, для каждого элемента `a1` из списка `comb` и для каждого возможного символа `a2` (цифра от 0 до 9) формируется число x по маске `31{a1}567{a2}`. Для каждого такого числа x выполняем проверку, является ли оно простым, с помощью функции `div(x)` (длина множества делителей числа x , `len(div(x))` равна 2). Если условие верно, то выводим на экран число x и произведение его чисел (результат работы функции `pr(x)`). Копируем в ответ все найденные пары чисел.

```
from itertools import *

def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

def pr(x):
    p = 1
    for d in str(x):
        p*= int(d)
    return p

comb = []
for l in 0,1:
    for x in product('0123456789', repeat=l):
        comb.append(''.join(x))

for a1 in comb:
    for a2 in '0123456789':
        x = int(f'31{a1}567{a2}')
        if len(div(x))==2:
            print(x, pr(x))
```

Ответ:

315671	630
315677	4410
3135677	13230
3145673	7560
3145679	22680
3165671	3780
3195679	51030

Telegram: @fast_ege

DZ_253_9

Задание №9(635)

Назовём нетривиальным делителем натурального числа его делитель, не равный единице и самому числу. Найдите все натуральные числа, принадлежащие отрезку $[106732567; 152673836]$ и имеющие ровно три нетривиальных делителя. Для каждого найденного числа запишите в ответе само число и его наибольший нетривиальный делитель. Найденные числа расположите в порядке возрастания. Например, для числа 2018 имеем следующие делители 2 и 1009. Поэтому результатом (не принимая во внимание количества делителей) будет пара чисел

2018 1009

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h21m40s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Поскольку нет необходимости считать 1 и само число (нам нужны только нетривиальные делители), будем перебирать все числа от 2 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d делитель i и результат деления $x//i$.

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 106732567 до 152673836 включительно (т.е. указываем диапазон $(106732567, 152673837)$). Для каждого числа из диапазона проверяем, является ли оно полным квадратом какого либо числа $((x**0.5)\%1==0)$ и существует ли у него ровно три делителя (длина множества делителей числа $\text{len}(\text{div}(x))$ равна трем). Выводим на экран число x и его наибольший нетривиальный делитель.

Копируем в ответ все найденные пары чисел.

```
def div(x):
    d = set()
    for i in range(2, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(106_732_567, 152_673_837):
    if (x**0.5)%1==0 and len(div(x))==3:
        print(x, max(div(x)))
```

Ответ:

112550881	1092727
131079601	1225043
141158161	1295029

DZ_253_10

Задание №10 (2592)

Найдите все натуральные числа, принадлежащие отрезку $[55\,000\,000; 60\,000\,000]$, у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа, справа от каждого числа запишите его наибольший нечётный делитель.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241284?t=0h24m25s

Решение

В первую очередь, создадим функцию $\text{div}(x)$, которая находит все делители числа x . Для этого создадим пустое множество d , в которое будем складывать делители. Затем перебираем все числа от 1 до \sqrt{x} (целая часть от корня числа). Если число x делится на i , то добавляем в множество d как делитель i , так и результат деления x/i .

Множество автоматически исключает дубли, и в конце возвращается отсортированный список делителей.

После того как у нас есть функция для нахождения делителей, мы перебираем все числа в диапазоне от 55000000 до 60000001. Возьмем число $x1$ и будем делить его на 2 до тех пор, пока оно делится на 2. То есть пока $x1\%2$ равно 0, пока число четное, мы будем его делить нацело. Таким образом мы оставляем только нечетный делитель, нечетное произведение простых множителей.

После этого проверяем, является ли число $x1$ полным квадратом и имеет ли это число ровно 5 делителей (длина множества делителей числа $x1$ равна 5). Если условия выполняются, то выводим на экран число x и его наибольший нечетный делитель ($\max(\text{div}(x1))$). Копируем в ответ все найденные пары чисел.

```
def div(x):
    d = set()
    for i in range(1, int(x**0.5)+1):
        if x%i==0:
            d.add(i)
            d.add(x//i)
    return sorted(d)

for x in range(55_000_000, 60_000_001):
    x1 = x
    while x1%2==0:
        x1 = x1 // 2
    if (x1**0.5)%1==0 and len(div(x1))==5:
        print(x, max(div(x1)))
```

Ответ:

55383364	13845841
56796482	28398241
58492928	28561
59105344	923521
59969536	14641
59973152	1874161