

Информатика. Щелчок 2022. Задание 25. Теория.

Задание 25 связано с обработкой чисел на отрезке, ключевое условие которых будет связано с делимостью (или неделимостью) на что-то.

Можно выделить три прототипа 25х задач:

- 1) Классические задачи на обработку всего отрезка
- 2) Задачи на оптимизацию (через основную теорему арифметики)
- 3) Маска (задача с досрока).

Примечание для всех задач:

Простое число — натуральное число, которое имеет ровно 2 натуральных делителя: 1 и само число.

Нетривиальные делители числа — все натуральные делители числа, кроме 1 и самого числа.

Но перед тем, как рассмотреть каждую из задач, научимся решать некоторые очень важные подзадачи:

Задача 1

Определите, является ли число 120 простым. Если да, выведите True, иначе False.

Ответ.

False

Решение.

Самой простой идеей является проверка того, делится ли число 120 на любое из чисел на отрезке $[2; 119]$. Если да, то значит число 120 не является простым (из определения выше).

Можно также оптимизировать программу, поскольку существует теорема, которая гласит о том, что если число не делится ни на один делитель от 2 до целой части корня самого числа включительно, то данное число является простым.

Ниже можно увидеть реализацию данной задачи

```
def prime(n):
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

print(count_div(120))
```

Итак, решим задачу следующего плана:

Задача 2

Найти количество натуральных делителей у числа 120, в том числе и тривиальные

Ответ.

16

Решение.

Воспользуемся идеей из предыдущего задания: проверим, если проверяемое число делится на отрезке от 2 до корня включительно, то существует сопряженное число этому числу, которое находится справа от корня. Если число является точным квадратом (например 25), то нужно проверить, не является ли сопряженный делитель равный найденному.

```
def count_div(n): # Счетчик всех делителей
    k = 2 # 1 и само число - делители числа всегда
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            k += 1
            if n // i != i:
                k += 1
    return k

print(count_div(120))
```

Если нам требуется найти все нетривиальные делители числа, то к делаем равным 0 (не учитываем 1 и само число).

И рассмотрим еще одну подзадачу:

Задача 3

Вывести в порядке возрастания все нетривиальные делители числа 49500.

Ответ.

2 3 4 5 6 9 10 11 12 15 18 20 22 25 30 33 36 44 45 50 55 60 66 75 90 99 100 110 125
132 150 165 180 198 220 225 250 275 300 330 375 396 450 495 500 550 660 750 825 900
990 1100 1125 1375 1500 1650 1980 2250 2475 2750 3300 4125 4500 4950 5500 8250 9900
12375 16500 24750

Решение.

```
def dividers(n): # Отсортированный массив нетривиальных делителей
    a = []
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            a.append(i)
            if n // i != i:
                a.append(n // i)
    a.sort()
    return a
```

Классические представители задач первого типа:

Задача 4

Найти количество простых чисел на промежутке [13123, 321321].

Ответ.

26152

Решение.

```
def prime(n): # Проверка на простоту
    if n == 1: return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True
```

```
counter = 0
for i in range(13123, 321322):
    if prime(i):
        counter += 1
print(counter)
```

Задача 5

Найти количество чисел из промежутка $[421431, 754123]$, у которых ровно 12 натуральных делителей.

Ответ.

29911

Решение.

```
def count_div(n): # Счетчик всех делителей
    k = 2 # 1 и само число - делители числа всегда
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            k += 1
            if n // i != i:
                k += 1
    return k

counter = 0
for i in range(421431, 754124):
    if count_div(i) == 12:
        counter += 1
print(counter)
```

Задача 6

Найти на промежутке $[21321, 99999]$ числа, у которых ровно 2 различных нетривиальных делителя. Найдите наибольшие 5 подходящих чисел и минимальный нетривиальный делитель каждого числа. В ответ запишите найденные 5 чисел в порядке убывания и делитель каждого из чисел в таком порядке:

Число1, Дел1; Число2, Дел2; и т.д. (После каждой запятой или точки с запятой ставится пробел, каждая группа (число и его минимальный делитель) отделяется от следующей группы точкой с запятой, после последней группы никакой знак не ставится).

Ответ.

99998, 2; 99993, 3; 99987, 3; 99986, 2; 99985, 5

Решение.

```
def count_div(n):
    k = 0 # считаем нетривиальные
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            k += 1
            if n // i != i:
                k += 1
    if k > 2:
        break
    return k

def min_divider(n):
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return i

counter = 0 # счетчик чисел
for i in range(99999, 21320, -1):
    if count_div(i) == 2:
        counter += 1
        print(i, min_divider(i), sep=', ', end='; ')
    if counter == 5:
        break
```

Рассмотрим задачи второго типа:

Их отличие от предыдущих в том, что мы рассматриваем числа, принадлежащие какому-то очень большому отрезку, что обычным перебором решить представляется невозможным. Рассмотрим на примере:

Задача 7

Найдите все натуральные числа, принадлежащие отрезку $[22\,222\,222; 88\,888\,888]$, у которых ровно 5 различных делителей. В ответе укажите количество данных чисел.

Ответ.

6

Решение.

Существует основная теорема арифметики — она говорит нам о том, что любое натуральное x можно представить в виде произведения простых чисел в какой либо неотрицательной целой степени. Так, например, $6 = 2^1 * 3^1$, $50 = 2^1 * 5^2$, $x = a_1^{p_1} * a_2^{p_2} * \dots * a_n^{p_n}$

Но нам важно следствие из нее — число делителей любого числа $k(x)$ можно представить в виде $k(x) = (p_1 + 1) * (p_2 + 1) * \dots * (p_n + 1)$.

По условию задачи мы знаем, что $k(x) = 5$. Исходя из выражения выше можем понять, что так как 5 простое число, то получить мы его могли только в случае, если только единственная скобка имела значение, равное 5, а все остальные 1. Отсюда следует, что у нас есть единственное ненулевое p и искомые нами числа имеют вид $x = a^4$, где a — простое число.

Программа ниже реализует знание об этом: мы находим простые числа на отрезке, после чего проверяем, являются ли они простыми и если да, то возводим в 4ую степень и проверяем принадлежность отрезку по условию задачи.

```
def prime(x):  
    for i in range(2, int(x**0.5)+1):  
        if x % i == 0:  
            return False  
    return True  
  
ans = 0  
for i in range(int(22222222**0.25), int((88888888+1)**0.25)+1):  
    if prime(i):  
        ans += 1  
print(ans)
```

Задача 8

Найдите все натуральные числа, принадлежащие отрезку $[33\,222\,555; 99\,888\,999]$, у которых ровно 3 различных нечётных делителя (количество чётных делителей может быть любым). В ответе укажите количество данных чисел.

Ответ.

1793

Решение.

Рассуждения точно такие же как и в предыдущей задаче — сначала понимаем, какие числа в принципе имеют 3 различных делителя. Это простые числа в квадрате. Поскольку по условию задачи требуется, чтобы было именно три различных нечетных делителя, а количество четных может быть любым, то нам подходят не только $x = a^2$, а в целом все $x = a^2 * 2^i$, главное чтобы искомые числа были в промежутке по условию задачи.

```
def is_prime(n):
    for j in range(2, int(n**0.5)+1):
        if n % j == 0:
            return False
    return True

def divs(n):
    a=[]
    for j in range(2,int(n**0.5)+1):
        if n%j==0:
            a.append(j)
            if n!=n//j:
                a.append(n//j)
    return a

ans = 0
for j in range(3, 10000):
    if is_prime(j):
        for i in range(26):
            if 33222555 <= 2**i*j**2 < 99888999:
                ans += 1

print(ans)
```

Задача 9

Найдите все натуральные числа, принадлежащие отрезку $[33\,333\,333; 99\,999\,999]$, у которых ровно 7 различных четных делителей. В ответе укажите количество данных чисел.

Ответ.

2

Решение.

Рассуждения точно такие же как и в предыдущей задаче — сначала понимаем, какие числа в принципе имеют 7 различных делителя. Поскольку по условию задачи требуется, чтобы было именно 7 различных четных делителя, то значит и количество нечетных должно быть равно 7, то нам подходят только $x = 2 * a^6$.

```
def prime(x):
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            return False
    return True

ans = 0
for i in range(int(33333333**(1/6)), int((99999999+1)**(1/6))):
    if prime(i):
        if 33333333 <= 2*(i**6) <= 99999999:
            ans += 1
print(ans)
```

И напоследок рассмотрим задачи третьего типа:

Примечание для последующих задач:

Назовём маской числовую последовательность, в которой так также могут встретиться следующие символы:

- символ "?" означает ровно одну произвольную цифру;
- символ "*" означает любую последовательность любой длины; в том числе и пустую.

Задача 10

Сколько существует натуральных чисел, соответствующих следующей маске 987?654?32 и делящихся на 88 без остатка. В ответ запишите в порядке возрастания найденные числа и соответствующее им частное от деления на 88.

Ответ.

9871654232 112177889

9873654032 112200614

9876654832 112234714

9878654632 112257439

Решение.

```
for a1 in ('0123456789'):
    for a2 in ('0123456789'):
        s='987'+a1+'654'+a2+'32'
        n=int(s)
        if n%88==0:
            print(n,n//88)
```

Задача 11

Сколько существует натуральных чисел, соответствующих следующей маске ???1??1? (число не может начинаться с нуля) и делящихся на 434343 без остатка. В ответ запишите в порядке возрастания найденные числа и соответствующее им частное от деления на 434343.

Ответ.

79919112 184

Решение.

```
for a1 in ('123456789'):
    for a2 in ('0123456789'):
        for a3 in ('0123456789'):
            for a4 in ('0123456789'):
                for a5 in ('0123456789'):
                    for a6 in ('0123456789'):
                        s=a1+a2+a3+'1'+a4+a5+'1'+a6
                        n=int(s)
                        if n%434343==0:
                            print(n,n//434343)
```

Задача 12

Сколько существует натуральных чисел, соответствующих следующей маске 123?34? и их сумма цифр делилась на 9. В ответ запишите количество таких чисел.

Ответ.

11

Решение.

```
counter=0
for a1 in ('0123456789'):
    for a2 in ('0123456789'):
        s='123'+a1+'34'+a2
        n=int(s)
        if n%9==0:
            counter+=1
print(counter)
```