

. Текстовый разбор домашки 1

DZ_25_1

Задача № 1 (3229)

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.

Среди натуральных чисел, не превышающих 10^9 , найдите все числа, соответствующие маске 12345?6?8 и делящиеся на 17 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им частные от деления на 17.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h0m0s

Решение

Запустим IDLE, будем использовать функцию `fnmatch` для работы с шаблонами строк. Мы будем перебирать все числа от 0 до 10^9 , которые делятся на 17.

Алгоритм:

1. Генерация чисел, кратных 17: Мы будем перебирать все числа от 0 до 10^9 , которые делятся на 17. Это можно сделать с шагом 17, начиная с 0.
2. Проверка соответствия маске: Для каждого числа будем проверять, соответствует ли оно маске '12345?6?8' с помощью функции `fnmatch`.
3. Вывод результата: Если число соответствует маске, выводим число и его частное при делении на 17.

```
from fnmatch import *

for x in range(0, 10**9, 17):
    if fnmatch(str(x), '12345?6?8'):
        print(x, x//17)
```

Ответ:

123450668	7261804
123451688	7261864
123456618	7262154
123457638	7262214
123458658	7262274
123459678	7262334

Telegram: @fast_ege

DZ_25_2

Задача №2 (4628)

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 $123*4?5$ соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^8 , найдите все числа, соответствующие маске $12*4?65$, делящиеся на 161 без остатка.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 161.

Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h2m30s

Решение

Запустим IDLE, будем использовать функцию `fnmatch` для работы с шаблонами строк. Мы будем перебирать все числа от 0 до 10^8 , которые делятся на 161, для этого в цикле используем шаг 161.

Алгоритм:

- Перебираем числа от 0 до 10^8 , деля их на 161.
- Для каждого числа проверяем, соответствует ли оно маске $12*4?65$ с помощью функции `fnmatch`.
- Если число подходит, выводим его и результат деления на 161.

```
from fnmatch import *

for x in range(0, 10**8, 161):
    if fnmatch(str(x), '12*4?65'):
        print(x, x//161)
```

Ответ:

1234065	7665
12004965	74565
12214265	75865
12294765	76365
12504065	77665
12584565	78165
12874365	79965
12954865	80465

Telegram: @fast_ege

DZ_25_3

Задача № 3 (5382)

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 $123*4?5$ соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^8 , найдите все числа, соответствующие маске $*1?542?$, делящиеся на 2084 без остатка.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 2084.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h4m10s

Решение

Запустим IDLE, будем использовать функцию `fnmatch` для работы с шаблонами строк. Мы будем перебирать все числа от 0 до 10^8 , которые делятся на 2084, для этого в цикле используем шаг 2084.

Алгоритм:

- Перебираем числа от 0 до 10^8 , с шагом 2084.
- Для каждого числа проверяем, соответствует ли оно маске $12*4?65$ с помощью функции `fnmatch`.
- Если число подходит, выводим его и результат деления на 2084.

```
from fnmatch import *

for x in range(0, 10**8, 2084):
    if fnmatch(str(x), '*1?542?'):
        print(x, x//2084)
```

Ответ:

22165424	10636
52135428	25017
65135420	31255
78185428	37517
91185420	43755
95105424	45636

Telegram: @fast_ege

DZ_25_4

Задача № 4 (9754)

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

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

Например, маске 1234?5 соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^8 , найдите все числа, соответствующие маске 3?157, делящиеся на 2023 без остатка.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 2023. Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h5m50s

Решение

Запустим IDLE, будем использовать функцию `fnmatch` для работы с шаблонами строк. Мы будем перебирать все числа от 0 до 10^8 , которые делятся на 2023, для этого в цикле используем шаг 2023.

Алгоритм:

- Перебираем числа от 0 до 10^8 , деля их на 2023.
- Для каждого числа проверяем, соответствует ли оно маске 3?157 с помощью функции `fnmatch`.
- Если число подходит, выводим его и результат деления на 2023.

```
from fnmatch import *

for x in range(0, 10**8, 2023):
    if fnmatch(str(x), '3?1*57'):
        print(x, x//2023)
```

Ответ:

321657	159
34105757	16859
35117257	17359
36128757	17859
37140257	18359
38151757	18859
39163257	19359

Telegram: @fast_ege

DZ_25_5

Задача №5 (12741)

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 1234?5 соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маскам $4*5*6$ и $?7468?$, делящиеся на 1234 без остатка.

В ответе запишите в первом столбце таблицы все найденные числа в порядке убывания, а во втором столбце – соответствующие им результаты деления этих чисел на 1234.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h6m55s

Решение

Будем использовать функцию `fnmatch` для работы с шаблонами строк. Мы будем перебирать все числа от 0 до 10^{10} , которые делятся на 1234, для этого в цикле используем шаг 1234. Наши числа должны соответствовать одновременно двум маскам, $4*5*6$ и $?7468?$. Из этих чисел мы сформируем список, который затем отсортируем по убыванию, и наконец выведем ответ.

Алгоритм:

- Создаем пустой список `a` для результатов..
- Перебираем числа от 0 до 10^{10} , деля их на 1234.
- Для каждого числа проверяем, соответствует ли оно маске одновременно двум маскам, $4*5*6$ и $?7468?$ с помощью функции `fnmatch`.
- Если число подходит, добавляем его в список `a`. Числа в списке располагаются в порядке возрастания.
- Сортируем список `a` в порядке убывания (переворачиваем его)
- Выводим найденные числа и соответствующие им результаты деления на 1234.

```
from fnmatch import *
```

```
a = []
```

```
for x in range(0,10**10,1234):
    if fnmatch(str(x),'4*5*6') and fnmatch(str(x),'?74*68?'):
        a.append(x)

a = a[::-1]

for x in a:
    print(x,x//1234)
```

Ответ:

4749516686	3848879
4747665686	3847379
4745814686	3845879
4745197686	3845379
4744580686	3844879
4741495686	3842379

Telegram: @fast__ege

DZ_25_6

Задание №6 (16270)

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.

Среди натуральных чисел, не превышающих 10^{13} , найдите все числа, соответствующие маске $12?345?67089?$, делящиеся на 206 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 206.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h10m35s

Решение

В данной задаче необходимо перебрать очень много чисел, поэтому перебором именно чисел и функцией `fnmatch` пользоваться неудобно. Для решения используется обычный перебор цифровых комбинаций.

Алгоритм:

- Создаем пустой список `a` для результатов.
- Перебираем все возможные комбинации цифр для `a1`, `a2` и `a3`: В коде используются три вложенных цикла для перебора цифр от 0 до 9. Это даёт 1000 возможных комбинаций (10 для каждой переменной), так как для каждой из переменных (`a1`, `a2`, `a3`) перебираются все 10 цифр.
- Для каждой комбинации значений `a1`, `a2` и `a3`, создаётся число с помощью строкового форматирования. Формируется число вида `12 a1 3452 a2 67089 a3`, где `a1`, `a2`, `a3` — это переменные, значения которых берутся из циклов. Эти символы объединяются в строку, которая потом преобразуется в целое число (`int`).

- Каждое сформированное число x проверяем на делимость на 206. Если число делится на 206 без остатка ($x \% 206 == 0$), то оно добавляется в список a .
- Сортируем список a в порядке возрастания.
- Выводим числа и результаты деления

```
a = []

for a1 in '0123456789':
    for a2 in '0123456789':
        for a3 in '0123456789':
            x = int(f'12{a1}345{a2}67089{a3}')
            if x%206 == 0:
                a.append(x)

a.sort()
for x in a:
    print(x, x//206)
```

Ответ:

1203458670898	5842032383
1223459670896	5939124616
1233450670896	5987624616
1253451670894	6084716849
1273452670892	6181809082
1293453670890	6278901315

Telegram: @fast_ege

DZ_25_7

Задание № 7(13832)

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

- символ «Ч» означает ровно одну произвольную чётную цифру;
- символ «Н» означает ровно одну произвольную нечётную цифру.

Например, маске НЧЗ соответствуют числа 723, 943, 183 и т.д. Среди натуральных чисел, не превышающих 10^9 , найдите все числа, соответствующие маске ЧНЧЧНЧН77, делящиеся на 7777 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 7777.

Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h13m20s

Решение

Для того, чтобы не писать огромное количество циклов, будем использовать функции для работы с итерациями, импортируем `itertools`.

Алгоритм:

- Создаём пустой список для результатов.

- С помощью функции `product` перебираем все возможные комбинации четырех символов из строки '02468' (т.е. всех четных цифр). Переменные `a1`, `a2`, `a3`, `a4` будут представлять каждую из этих цифр в комбинации.
- Перебираем все возможные комбинации трех символов из строки '13579' (т.е. всех нечетных цифр). Переменные `b1`, `b2`, `b3` будут представлять эти цифры в комбинации.
- Формируем число `x`: сначала добавляются значения переменных `a1`, `b1`, `a2`, `a3`, `b2`, `a4`, `b3`, в конце - две цифры 77. Это символы объединяются в строку, которая затем преобразуется в целое число.
- Проверяем число `x` на соответствие условиям задачи – число должно начинаться со значимой цифры (не нуля) и делиться на 7777. Если число соответствует, то оно добавляется в список `a`.
- Сортируем список `a` в порядке возрастания.
- Выводим числа и результаты деления

```
from itertools import *

a = []

for a1,a2,a3,a4 in product('02468',repeat=4):
    for b1,b2,b3 in product('13579',repeat=3):
        x = int(f'{a1}{b1}{a2}{a3}{b2}{a4}{b3}77')
        if a1!='0' and x%7777==0:
            a.append(x)

a.sort()
for x in a:
    print(x, x//7777)
```

Ответ:

214652977	27601
416854977	53601
418410377	53801
438630577	56401
458850777	59001
634610977	81601
836812977	107601

Telegram: @fast_ege

DZ_25_8

Задание №8 (14435)

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

- символ «`?`» означает ровно одну произвольную цифру;
- символ «`#`» означает последовательность из ровно трёх нечётных цифр.

Например, маске `123#4?5` соответствуют числа `123333405` и `123171405`. Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске `71#39?28`, делящиеся на 2024 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке

возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 2024. Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h17m20s

Решение

Будем использовать функции для работы с итерациями, импортируем itertools.

Алгоритм:

- Создаём пустой список `a` для результатов.
- Для переменной `a1` перебираем все цифры от 0 до 9 (т.е. все символы строки '123456789')
- Перебираем все возможные комбинации трех символов из строки '13579' (т.е. всех нечетных цифр). Переменные `b1`, `b2`, `b3` будут представлять эти цифры в комбинации.
- Формируем число `x`: добавляем значения переменных в шаблон '71 b1b2b3 39 a1 28'. Это символы объединяются в строку, которая затем преобразуется в целое число.
- Проверяем число `x` на делимость на число 2024. Если число соответствует, то оно добавляется в список `a`.
- Сортируем список `a` в порядке возрастания.
- Выводим числа и результаты деления

```
from itertools import *

a = []

for a1 in '0123456789':
    for b1,b2,b3 in product('13579',repeat=3):
        x = int(f'71{b1+b2+b3}39{a1}28')
        if x%2024==0:
            a.append(x)

a.sort()
for x in a:
    print(x,x//2024)
```

Ответ:

7113139528	3514397
7117339328	3516472
7155339928	3535247
7159539728	3537322

Telegram: @fast_ege

DZ_25_9

Задание №9(18298)

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

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

Например, маске 123#4?5 соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1592\#6?8$ и делящиеся на 1996 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 1996. Количество строк в таблице для ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h20m20s

Решение

В первую очередь оценим максимальную длину последовательности «#». Для этого сравним маску числа с 10^{10} . Очевидно, что последовательность «#» не может быть длиннее трёх символов, в противном случае число $1592\#6?8$ будет превышать 10^{10} . Будем использовать функции для работы с итерациями, импортируем `itertools`.

Алгоритм:

- Создаём пустой список `comb` для последовательности «#».
- С помощью функции `product` генерируем все возможные комбинации из 0-4 символов четных чисел.
- Каждую комбинацию мы преобразуем в строку с помощью `''.join(x)` и добавляем её в список `comb`.
- Создаём множество `a`, в котором будут храниться числа, удовлетворяющие некоторому условию.
- Перебираем все цифры от 0 до 9 (переменная `a1`).
- Для каждой строки из списка `comb` (переменная `b1`), генерируем число вида: `'1592{b1}6{a1}8'`, где `b1` — это комбинация из чётных цифр, а `a1` — цифра от 0 до 9. Получившуюся строку преобразуем в число с помощью `int()`.
- Проверяем, делится ли это число на 1996. Если да, добавляем его в множество `a`. Сортируем список `a` в порядке возрастания.
- Выводим числа и результаты деления.

```
from itertools import *

comb = []
for i in range(4):
    for x in product('02468', repeat=i):
        comb.append(''.join(x))

a = set()
for a1 in '0123456789':
    for b1 in comb:
        x = int(f'1592{b1}6{a1}8')
        if x % 1996 == 0:
            a.add(x)

a = sorted(a)
for x in a:
    print(x, x // 1996)
```

Ответ:

1592464688	797828
1592484648	797838

Задание №10 (3376)

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.

Среди натуральных чисел, не превышающих 10^9 , найдите все числа, соответствующие маске $1*5*9$, значения разрядов в которых идут в строго возрастающем порядке, и делящиеся на 21 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им частные от деления на 21. Количество строк для ввода ответа избыточно.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241267?t=0h24m35s

Решение

Оценим максимальную длину последовательностей «*». Для этого сравним маску числа с 10^9 . Очевидно, что последовательность «#» не может быть длиннее шести символов, в противном случае число $1*5*9$ будет превышать 10^9 . Будем использовать функции для работы с итерациями, импортируем `itertools`.

Алгоритм:

- Создаём пустой список `comb`.
- С помощью функции `product` генерируем все возможные комбинации из 0-6 цифр.
- Каждую комбинацию мы преобразуем в строку с помощью `''.join(x)` и добавляем её в список `comb`.
- Создаём список `a`, в котором будут храниться числа.
- Для каждой пары строк `a1` и `a2` из списка `comb` формируем строку `s` вида `1 a1 5 a2 9`. Преобразуем строку в число с помощью `int()`.
- Проверяем, соответствует ли полученное число условиям задачи. Если число больше 10^9 , то оно не подходит, дальнейшие проверки для текущей комбинации не проводятся. Если число меньше, делится на 21 и его цифры идут в строго возрастающем порядке (каждая цифра меньше следующей), то это число добавляется в список `a`.
- Сортируем список `a` в порядке возрастания.
- Выводим числа и результаты деления.

```
from itertools import *

comb = []
for l in range(7):
    for x in product('0123456789', repeat=l):
        comb.append(''.join(x))

a = []
for a1 in comb:
    for a2 in comb:
        s = f'1{a1}5{a2}9'
        x = int(s)
        if x > 10**9: break
        if x % 21 == 0 and all(s[i] < s[i+1] for i in range(len(s)-1)):
            a.append(x)
```

```
a.sort()
for x in a:
    print(x,x//21)
```

Ответ:

12579	599
123459	5879
134589	6409
1234569	58789
1356789	64609

Telegram: @fast_ege