Конспект 1

Strim_25_1

Что такое файловая маска и модуль fnmatch

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=0h1m50s

Файловый шаблон представляет собой специальное выражение, предназначенное для идентификации группы файлов, имеющих общую часть в своих именах.
Для создания шаблонов используются обычные символы (буквы, цифры и другие знаки), а также два специальных символа: звездочка (*) и вопросительный знак (?). Звездочка обозначает любую последовательность символов произвольной длины, включая пустую строку. Вопросительный знак соответствует любому одиночному символу.

Эти обозначения также применяются в задаче № 25, где требуется проверка соответствия строк определенному шаблону. В Python существует встроенный модуль fnmatch, предназначенный для проверки соответствия имен файлов указанным шаблонам. Этот модуль использует тот же синтаксис, согласно которому звездочка соответствует любой последовательности символов, а вопросительный знак – одному произвольному символу.

Задача № 1 (7942)

(ЕГКР-2024) 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 12300425.
Среди натуральных чисел, не превышающих 10^10, найдите все числа, соответствующие маске 54?1?3*7, делящиеся на 18579 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 18579.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=0h4m55s

Решение

Шаги решения:
Перебор чисел: Так как условие требует нахождения чисел, делящихся на 18579, имеет смысл начать с поиска таких чисел. Для этого можно использовать цикл, начиная с 0 и добавляя каждый раз 18579, пока результат не превышает 10^10. Почему начинаем с 0? Потому что 0 — это универсальное число, которое делится на любое другое целое число без остатка. Это означает, что мы перебираем только те числа, которые гарантированно делятся на 18579.

Проверка соответствия маске: После того как найдено число, делящееся на 18579, необходимо проверить, соответствует ли оно маске 54?1?3*7. Для этого используется модуль fnmatch. Этот модуль предоставляет функцию fnmatch(), которая сравнивает строку с шаблоном (маской). Маска использует символы ? для обозначения одного произвольного символа и * для обозначения любого количества символов. Функция fnmatch() принимает две строки: само число (преобразованное в строку через str(x)), и шаблон-маску ('54?1?3*7'). Если число соответствует маске, функция возвращает True.

Вывод результата: Если число удовлетворяет всем условиям, выводим его и результат деления на 18579.

```
# Импорт модуля fnmatch
from fnmatch import *
# Цикл по числам, кратным 18579
for x in range(0, 10**10, 18579):
    # Проверка соответствия маске
    if fnmatch(str(x), '54?1?3*7'):
        # Вывод результатов
        print(x, x//18579)
```

Ответ:

545163597	29343
5411932647	291293
5421036357	291783
5451134337	293403
5461538577	293963
5481232317	295023
5491636557	295583

Задача №2 (6788)

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 12300425.
Найдите все числа, меньшие 10^8, соответствующие маске 1*2??76 и делящиеся без остатка на 1923. В качестве ответа приведите все найденные числа в порядке возрастания, справа от каждого числа выведите результат его деления на 1923.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=0h15m25s

Решение

Решим эту задачу аналогичным образом: нужно перебрать все числа, которые делятся на 1923, и проверить, соответствуют ли они маске 1*2??76. Если число проходит проверку, вывести его и результат деления на 1923.
Алгоритм решения:
1. Перебор чисел: Начинаем с 0, так как это число делится на любое другое число без остатка, и увеличиваем значение с шагом 1923, чтобы получить числа, кратные 1923.
2. Проверка соответствия маске: Используем модуль fnmatch, который позволяет проверять соответствие строки определенному шаблону (маске). Преобразуем число x в строку с помощью функции str(), так как сравнение происходит именно со строками. Затем вызываем функцию fnmatch(), передавая ей строку и маску. Если строка соответствует маске, функция вернет True.
3. Вывод результата: Если число соответствует маске, выводим его и результат деления на 1923.

```
from fnmatch import *

for x in range(0,10**8,1923):
    if fnmatch(str(x),'1*2??76'):
        print(x, x//1923)
```

Ответ:

10022676	5212
12522576	6512
15022476	7812
17522376	9112
19829976	10312

Задача № 3 (6039)

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?58*129, которые делятся без остатка только на одно из чисел 117, 119, 121.

В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, справа от каждого числа запишите результат его деления на то из чисел 117, 119, 121, на которое это число делится без остатка.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=0h24m10s

Решение

1. Перебор чисел: Создадим три цикла, каждый из которых будет перебирать числа, кратные одному из указанных делителей (117, 119, 121).
2. Проверка маски: Для каждого числа проверим, соответствует ли оно маске 1?58*129.
3. Проверка делимости: Убедимся, что число делится только на одно из указанных чисел.
4. Сбор результатов: Соберём все подходящие числа в список и отсортируем его.
5. Вывод результата: Выведем числа в порядке возрастания вместе с результатом деления на соответствующее число.

```
from fnmatch import *
# Список для хранения результатов
ans = []
# Перебор чисел кратных 117
for x in range(0,10**8,117):
    if fnmatch(str(x),'1?58*129') and x%119!=0 and x%121!=0:
        ans.append([x,x//117])
        #print(x,x//117)
# Перебор чисел кратных 119
for x in range(0,10**8,119):
    if fnmatch(str(x),'1?58*129') and x%117!=0 and x%121!=0:
        ans.append([x,x//119])
        #print(x,x//119)
# Перебор чисел кратных 121
for x in range(0,10**8,121):
    if fnmatch(str(x),'1?58*129') and x%117!=0 and x%119!=0:
        ans.append([x,x//121])
        #print(x,x//121)
#ans.sort()
# Сортируем результаты по возрастанию первого элемента (числа)
ans = sorted(ans)
# Выводим
for x in ans:
    print(*x)
```

Ответ:

10581129	90437
12589129	105791
13582129	112249
17587129	147791

* Метод .sort() сортирует уже созданный список, а функция sorted() создает новый сортированный список.

Задача № 4 (4987)

(А. Кабанов) 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 12300425.
Среди натуральных чисел, не превышающих 10^9, найдите все числа, соответствующие маске 1?34567?9 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_456241266?t=0h40m40s

Решение

Маска имеет вид 1?34567?9, где знаки вопроса могут принимать значения от 0 до 9. Нам нужно проверить, делится ли каждое число, соответствующее маске, на 17 без остатка. Для каждого знака вопроса будем перебирать все возможные значения от 0 до 9.
Таким образом:
Внешний цикл for a1 in '0123456789': переберёт все возможные значения первой неизвестной цифры маски от 0 до 9.
Внутренний цикл for a2 in '0123456789': переберёт все возможные значения второй неизвестной цифры маски также от 0 до 9.
Строка x = int(f'1{a1}34567{a2}9') сформирует число, подставляя текущие значения a1 и a2 вместо знаков вопроса в маске.
Условие if x % 17 == 0: проверит, делится ли сформированное число на 17 без остатка.
Если условие выполняется, программа выводит само число и результат целочисленного деления этого числа на 17.
Так мы найдём все числа, удовлетворяющие условиям задачи, и напечатаем их вместе с соответствующими частными от деления на 17.

```
for a1 in '0123456789':
```

```
    for a2 in '0123456789':
        x = int(f'1{a1}34567{a2}9')
        if x%17==0:
            print(x,x//17)
```

Ответ:

113456759	6673927
133456749	7850397
153456739	9026867
173456729	10203337
193456719	11379807

Задача №5 (4988)

(А. Кабанов) 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 12300425.
Среди натуральных чисел, не превышающих 10^9, найдите все числа, соответствующие маске 123*567? и делящиеся на 169 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 169.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=0h48m15s

Решение

Чем данная задача отличается от предыдущих? Тем, что у нас есть маска, содержащая один вопросительный знак и одну звездочку. Звездочка обозначает любую последовательность цифр, включая пустую (то есть длину последовательности 0). Таким образом, звездочка представляет собой некоторое количество произвольных цифр длиной от 0 до бесконечности.
Теперь рассмотрим ограничение: натуральные числа должны быть меньше или равны10^9. Какое максимальное количество цифр может быть в нашем числе? Из условия следует, что оно должно содержать не более девяти цифр.
Далее обратимся к нашей маске. Уже известно, что в ней присутствуют семь цифр (1234567), следовательно, в звездочке может находиться от 0 до 2 цифр. Если бы в звездочке было три цифры, получилось бы десятизначное число, которое превышает допустимый предел (10^9). Значит, максимальная длина звездочки – две цифры.

Перебор вариантов
Как организовать перебор? Для вопроса все понятно: он заменяется любой цифрой от 0 до 9. Но как перебирать звездочку? Лучше всего воспользоваться библиотекой itertools, которая позволяет эффективно генерировать различные комбинации цифр.
Создадим список возможных комбинаций для звездочки. Будем перебирать их по длине от 0 до 2, а также генерировать все возможные комбинации цифр для каждой длины.

```
from itertools import *
# Создаем список возможных комбинаций для звёздочки
comb = []
#перебираем разные варианты длины от 0 до 2 цифр
for l in range(0,3):
    #перебираем комбинации цифр нужной длины
    for x in product('0123456789',repeat=l):
        comb.append(''.join(x))
#перебираем все комбинации для звездочки
for a1 in comb:
#перебираем все возможные значения для вопросика
for a2 in '0123456789':
     # формируем число согласно маске
      x = int(f'123{a1}567{a2}')
        if x%169==0:
            print(x,x//169)
```

Ответ:

12325677	72933
12385672	73288
123165679	728791
123225674	729146
123515678	730862
123575673	731217
123865677	732933
123925672	733288

Задание №6 (4989)

(А. Кабанов) 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 12300425.
Среди натуральных чисел, не превышающих 10^6, найдите все числа, соответствующие маске 12*45* и делящиеся на число 51 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 51.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=1h1m25s

Решение

Для начала разберемся с маской 12*45*. Звездочки означают, что на их месте могут находиться любые цифры, причем каждая звездочка может представлять собой одну или две цифры. Таким образом, максимальное количество цифр в числе, соответствующем данной маске, составляет шесть (поскольку одна цифра уже занята цифрой '1', другая – цифрой '2', третья – '4', четвертая – '5').
Мы можем варьировать длину цифр под звездочками от 0 до 2, чтобы получить все возможные комбинации.
Для этого воспользуемся модулем itertools, который позволяет генерировать всевозможные комбинации цифр для каждой звездочки. Мы создадим список всех возможных комбинаций длины от 0 до 2 для каждой звездочки, а затем сформируем числа, соответствующие нашей маске.

```
from itertools import product

# Создаем список всех возможных комбинаций цифр для каждой звездочки
comb = []
for l in range(0, 3):
    for x in product('0123456789', repeat=l):
        comb.append(''.join(x))
```

Теперь, когда у нас есть все возможные комбинации для звездочек, мы можем начать формировать числа, соответствующие маске 12*45*.

```
ans = []
for a1 in comb:
    for a2 in comb:
        # Формируем число согласно маске
        x = int(f'12{a1}45{a2}')
        # Проверяем, что число не превышает 10^6
        if x > 10**6:
            break
        # Проверяем делимость на 51
        if x % 51 == 0:
            # Добавляем пару [число, частное от деления на 51]
            ans.append([x, x // 51])
```

После того как мы собрали все подходящие числа, необходимо отсортировать их по возрастанию.

```
# Сортируем список пар по первому элементу (самому числу)
ans.sort()
```

```
# Выводим результаты
for x in ans:
    print(*x)
```

Таким образом, программа сначала генерирует все возможные комбинации цифр для каждой звездочки, затем проверяет каждое полученное число на соответствие условиям задачи (не превышает 10^6 и делится на 51), после чего сортирует результаты и выводит их в виде таблицы.

Важные моменты.
1. Контроль длины числа: Поскольку маска допускает наличие до двух цифр на каждую звездочку, важно контролировать общую длину числа, чтобы оно не превышало шести цифр.
2. Делимость на 51: После формирования числа обязательно проверять, делится ли оно на 51 без остатка.
3. Отсев больших чисел: Для ускорения работы программы сразу отбрасываются числа, которые больше 10^6.
4. Сортировка результатов: Чтобы вывести числа в порядке возрастания, используется сортировка списка пар по первому элементу (самому числу).
Вся программа целиком:

```
from itertools import *
comb = []
for l in range(0,3):
    for x in product('0123456789',repeat=l):
        comb.append(''.join(x))
ans = []
for a1 in comb:
    for a2 in comb:
        x = int(f'12{a1}45{a2}')
        if x>10**6: break
        if x%51==0:
            ans.append([x,x//51])
ans.sort()
for x in ans:
    print(*x)
```

Ответ:

122145	2395
122451	2401
124542	2442
124593	2443
127245	2495

Задание №7(6655)

(Е. Джобс) 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 12300425.

Найдите все числа, меньшие 10^12, соответствующие маске 123?4*5679 и делящиеся без остатка на 4013. В качестве ответа приведите все найденные числа в порядке возрастания, справа от каждого числа выведите результат его деления на 4013.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=1h12m30s

Проверка чисел fnmatch IV.
Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=1h22m5s

Решение

В данной задаче необходимо найти числа, удовлетворяющие шаблону 123?4*5679 и делящиеся на 4013 без остатка. Для этого требуется перебирать большое количество чисел – около 100 миллионов, что делает использование функции fnmatch неэффективным. Поэтому приходится прибегать к перебору различных комбинаций или ожидать длительное время, пока все числа будут проверены. Максимальная длина числа составляет 12 цифр, а символ «звездочка» (*) может иметь длину от 0 до 3 цифр. Мы будем генерировать все возможные комбинации для звездочки, используя модуль itertools. После этого соберем полное число и проверим, делится ли оно на 4013.

```
from itertools import product

# Генерация всех возможных вариантов для звездочки
comb = []
for l in range(0, 4):
    for x in product('0123456789', repeat=l):
        comb.append(''.join(x))

# Перебор всех возможных значений для первой цифры после 123
for a1 in '0123456789':
    # Перебор всех возможных значений для звездочки
    for a2 in comb:
        # Формирование полного числа
        x = int(f'123{a1}4{a2}5679')

        # Проверка деления на 4013
        if x % 4013 == 0:
            print(x, x // 4013)

#II способ
from fnmatch import *

for x in range(0,10**12, 4013):
    if fnmatch(str(x),'123?4*5679'):
        print(x,x//4013)
```

Ответ:

123240365679	30710283
123441015679	30760283
123641665679	30810283
123842315679	30860283

Задание № 8(7083)

Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:
— символ «A» означает ровно одну произвольную чётную цифру;
— символ «B» означает любую последовательность нечётных цифр произвольной длины; в том числе «B» может задавать и пустую последовательность.
Например, маске 123B4A5 соответствуют числа 123405 и 12399405.
Среди натуральных чисел, не превышающих 10^10, найдите все числа, соответствующие маске 1A2157B4, делящиеся на 133 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 133.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=1h24m45s

Решение

Маска имеет вид 1A2157B4, где:
- A — одна четная цифра,
- B — любая последовательность нечетных цифр длиной от 0 до 3.

Для начала рассмотрим все возможные комбинации для последовательности B. Поскольку она может содержать от 0 до 3 нечетных цифр, необходимо перебрать все такие комбинации.

```
from itertools import *
# Генерация всех возможных значений для B
comb = []
for l in range(0, 4):
    for x in product('13579', repeat=l):
        comb.append(''.join(x))
```

Теперь, когда у нас есть все возможные значения для B, мы можем генерировать числа, соответствующие маске, и проверять их делимость на 133.

```
# Перебираем все возможные значения A и B
ans = []
for a in '02468':
    for b in comb:
        # Формируем число согласно маске
        x = int(f'1{a}2157{b}4')
        # Проверяем условие делимости на 133
        if x % 133 == 0:
            ans.append([x, x // 133])
```

После того как мы собрали все подходящие числа, сортируем их по возрастанию и выводим результат.

```
# Сортируем результаты
ans.sort()
# Выводим ответы
for x in ans:
    print(*x)
```

Весь код решения задачи:

```
from itertools import *
comb = []
for l in range(0,4):
    for x in product('13579',repeat=l):
        comb.append(''.join(x))
ans = []
for a in '02468':
    for b in comb:
        x = int(f'1{a}2157{b}4')
        if x%133==0:
            ans.append([x,x//133])
ans.sort()
for x in ans:
    print(*x)
```

Ответ:

122157574	918478
1021575394	7681018
1421575554	10688538
1821575714	13696058

Задание №9 (5618)

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

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

Например, маске *ЧН2 соответствуют числа 7232, 612, 444692 и т.д.
Среди натуральных чисел, не превышающих 10^8, найдите все числа, соответствующие маске 123*НЧ56, 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_456241266?t=1h34m15s

Решение

Для начала определим значения символов маски:

- Ч — произвольная четная цифра,
- Н — произвольная нечетная цифра,
- звёздочка (*) — любая последовательность цифр, включая пустую.

Задача состоит в том, чтобы найти все числа, не превышающие 108108, которые соответствуют маске 123*НЧ56 и делятся на 206 без остатка. Затем нужно записать эти числа во второй столбец, соответствующий результату деления этих чисел на 206.

Рассмотрим структуру числа согласно маске:

1. Число начинается с фиксированных цифр 123.
2. Далее следует звёздочка (*), которая может представлять собой либо пустое множество, либо одну произвольную цифру.
3. После этого идёт нечетная цифра (Н).
4. Потом идет четная цифра (Ч).
5. Завершается число последовательностью 56.

Таким образом, структура числа выглядит следующим образом: 123[звёздочка][нечетная цифра][четная цифра]56. Важно отметить, что звёздочка может быть пустой или содержать одну цифру, поэтому общее количество цифр в числе варьируется от 7 до 8.

Теперь перейдем к решению задачи. Мы будем генерировать все возможные комбинации чисел, соответствующих данной маске, и проверять их делимость на 206.

```
from itertools import product

# Генерируем все возможные варианты для звездочки
comb = []
for l in 0, 1:
    for x in product('0123456789', repeat=l):
        comb.append(''.join(x))

# Перебираем все возможные комбинации
for a1 in comb:
    for a2 in '13579':  # Нечетные цифры
        for a3 in '02468':  # Четные цифры
            # Формируем число по шаблону
            x = int(f'123{a1}{a2}{a3}56')
            # Проверяем условие делимости на 206
            if x % 206 == 0:
                print(x, x // 206)
```

Результат работы программы:

Ответ:

1231056	5976
12313856	59776
12355056	59976
12375656	60076

Задание №10 (5033)

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 12300425.
Найдите все натуральные числа, делящиеся нацело на 114_8, восьмеричный код которых соответствует маске 1?345?700.
В ответе запишите найденные числа в десятичной системе счисления в порядке убывания, а справа от каждого числа – соответствующее частное от деления на 114_8.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241266?t=1h39m50s

Решение

Для начала обратим внимание на задачу: нам необходимо найти все натуральные числа, которые делятся на 114 в восьмеричной системе счисления, при этом их восьмеричный код должен соответствовать маске 1?345?700. Важно отметить, что числа должны быть представлены в восьмеричной системе, но результат требуется записать в десятичной системе.
Шаги решения
1. Перебор возможных значений:
2. Маска содержит два неизвестных символа (?), каждый из которых может принимать значения от 0 до 7 (так как система счисления восьмеричная). Это означает, что нам нужно перебрать все возможные комбинации этих символов.
3. Проверка условия делимости:
4. Для каждой комбинации проверяем, делится ли полученное число на 114 в восьмеричной системе. Для этого переводим число из восьмеричного представления в десятичное и делим его на 114, переведённое также в десятичную систему.
5. Формирование ответа:
6. Если условие выполняется, выводим само число в десятичном представлении и частное от деления на 114.

```
# Первый знак маски
for a1 in '01234567':
    # Второй знак маски
    for a2 in '01234567':
        # Формируем число в восьмеричной системе
        x = int(f'1{a1}345{a2}700', 8)
        # Проверяем деление на 114 в восьмеричной системе
        if x % int('114', 8) == 0:
            # Выводим число в десятичной системе и частное от деления
            print(x, x // int('114', 8))
```

Ответ :

21913536	288336
26106304	343504
30299072	398672
21913536	288336