

Конспект 1

Strim_17_1

Задание 17 о том, что дана некоторая последовательность чисел. Даны числа, записанные в текстовый файл. И нужно в этих числах что-то найти. Как правило, посчитать пары или тройки соседних элементов, которые подходят под какое-то условие.

Начнем с того, как организован перебор элементов в списке. Возьмём произвольный список чисел. Например:

$a = [5, 100, -7, 9, 4, 1]$.

Есть два способа перебора элементов списка.

Однако в этом задании, как правило требуется организовать не перебор чисел, а перебор соседних пар, соседних троек, соседних четверок.

Пара представляет собой пару последовательных элементов. В данном случае речь идет о переборе пар смежных чисел. Под парой подразумевается сочетание двух соседних элементов, расположенных рядом друг с другом.

Методы перебора

Рассмотрим методы перебора таких пар. Существуют два основных подхода. Первый заключается в переборе смежных пар, то есть тех, у которых порядковые номера отличаются на единицу (например, 0-1, 1-2, 2-3, 3-4, 4-5). Эти элементы являются соседями, так как разница между их индексами составляет ровно одну позицию. Следует отметить, что количество пар всегда на единицу меньше общего количества элементов. Например, если имеется шесть чисел, то образуется пять пар. Последнее число не имеет пары, так как отсутствует следующий элемент. При переборе пар по индексам необходимо останавливаться на предпоследнем индексе ($\text{len}(a) - 1$).

Второй подход предполагает использование функции `zip`. Он также позволяет получить пары смежных элементов. Функция `zip` принимает два списка: исходный список и тот же список, начиная со второго элемента (`a[1:]`), и последовательно сопоставляет их элементы. Это позволяет эффективно перебирать пары смежных элементов. Последний элемент не включается в пары, так как ему не соответствует последующий элемент. Таким образом, функция `zip` объединяет два списка параллельно и возвращает их элементы попарно. Данный метод подходит не только для пар, но и для других комбинаций элементов. Важно понимать, что мы рассматриваем исключительно перебор смежных пар, а не всех возможных сочетаний.

Задача № 1 (4301)

В файле [17-3.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -10 000 до 10 000 включительно. Определите и запишите в ответе сначала количество пар элементов последовательности, произведение которых положительно, а сумма кратна 7, затем минимальное из произведений элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=0h15m15s

Решение

Для начала необходимо открыть файл 17-3.txt, который содержит последовательность целых чисел. Эти числа могут находиться в диапазоне от -10000 до 10000 включительно. Задача состоит в том, чтобы определить количество пар элементов, удовлетворяющих двум условиям: произведение элементов должно быть положительным, а их сумма должна быть кратной 7. Под парой понимается два последовательных элемента списка. Также требуется найти минимальное значение произведения таких пар.

Чтобы решить задачу, начнем с чтения содержимого файла и преобразования строк в целые числа. Это позволит создать список чисел, с которым мы сможем работать далее.

```
a = [int(x) for x in open('17-3.txt')]
```

Теперь нам нужно пройти по списку и рассмотреть все пары последовательных элементов. Для этого используем функцию `zip()`, которая поможет соединить текущий элемент с последующим:

```
for x, y in zip(a, a[1:]):
```

Для каждой такой пары проверим выполнение двух условий:

1. Произведение элементов должно быть положительным ($x * y > 0$).
2. Сумма элементов должна быть кратной 7 ($(x + y) \% 7 == 0$).

Если оба условия выполняются, добавим произведение этой пары в список `ans`.

```
if x * y > 0 and (x + y) % 7 == 0:  
    ans.append(x * y)
```

После того как все пары будут обработаны, вычислим длину списка `ans`. Она покажет количество подходящих пар. Затем найдем минимальное произведение среди тех, что были сохранены в списке `ans`.

```
print(len(ans), min(ans))
```

Таким образом, программа находит количество пар, соответствующих заданным условиям, и минимальное произведение среди них.

Программа целиком:

```
a = [int(x) for x in open('17-3.txt')]  
ans = []  
for x, y in zip(a, a[1:]):  
    if x*y>0 and (x+y)%7==0:  
        ans.append(x*y)  
print(len(ans), min(ans))
```

Ответ:

359 115022

Задача №2 (4305)

В файле [17-3.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от -10 000 до 10 000 включительно. Определите и запишите в ответе сначала количество троек элементов последовательности, в которых произведение кратно 7, а сумма оканчивается на 5, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=0h23m45s

Решение

Важным нюансом этой задачи является проверка окончания суммы на 5.

Для проверки того, оканчивается ли сумма на 5, необходимо обратить внимание на следующий момент: стандартная операция взятия остатка от деления на 10 ($\% 10$) корректно определяет последнюю цифру числа только для неотрицательных значений. Например, для числа 137 остаток от деления на 10 равен 7, что соответствует последней цифре числа. Однако для отрицательных чисел ситуация меняется. Так, для числа -11, остаток от деления на 10 будет равен 9, что не является последней цифрой числа. Поэтому при работе с отрицательными числами необходимо использовать абсолютное значение числа, чтобы корректно проверить последнюю цифру.

Таким образом, для проверки условия "сумма оканчивается на 5" следует использовать выражение $\text{abs}(x + y + z) \% 10 == 5$.

Алгоритм решения

1. Считываем данные из файла и преобразуем их в список целых чисел.
2. Используем функцию `zip()` для создания троек последовательных элементов списка.
3. Для каждой тройки проверяем два условия:
 - о Произведение элементов тройки кратно 7: $x * y * z \% 7 == 0$
 - о Сумма элементов тройки оканчивается на 5: $\text{abs}(x + y + z) \% 10 == 5$
4. Если оба условия выполняются, добавляем сумму элементов тройки в список `ans`.
5. После обработки всех троек выводим количество подходящих троек и максимальную из их сумм.

```
# Чтение данных из файла и преобразование в список целых чисел
a = [int(x) for x in open('17-3.txt')]
# Список для хранения сумм подходящих троек
ans = []
# Перебор троек элементов с помощью функции zip()
for x, y, z in zip(a, a[1:], a[2:]):
    # Проверка условий
    if x * y * z % 7 == 0 and abs(x + y + z) % 10 == 5:
        # Добавление суммы в список
        ans.append(x + y + z)
# Вывод количества подходящих троек и максимальной из их сумм
print(len(ans), max(ans))
```

Запустив этот код, мы получим количество троек, удовлетворяющих условиям, и максимальную сумму среди них.

Ответ:
153 19285

Задача № 3 (4538)

В файле [17-205.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Определите количество пар, в которых хотя бы один из двух элементов заканчивается на 7, а их сумма делится на 12. В ответе запишите два числа: сначала количество найденных пар, а затем – максимальную сумму элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=0h33m20s

Решение

Важно учитывать следующие моменты при написании кода:

1. Условие, связанное с окончанием одного из элементов пары на 7, должно быть заключено в скобки, поскольку оно состоит из двух логических условий, объединенных оператором `or`. Это позволит Python корректно интерпретировать данное условие.
2. Проверку на делимость суммы на 12 следует выполнять без использования функции `abs()`, так как остаток от деления отрицательного числа также может быть равен нулю, что означает, что число делится на заданный делитель.

Теперь перейдем к коду:

```
# Чтение данных из файла
a = [int(x) for x in open('17-205.txt')]
# Список для хранения подходящих сумм
ans = []
# Перебор всех пар последовательных элементов
for x, y in zip(a, a[1:]):
    # Проверка обоих условий: наличие цифры 7 в конце хотя бы одного числа и делимость
    # суммы на 12
    if (abs(x) % 10 == 7 or abs(y) % 10 == 7) and (x + y) % 12 == 0:
        # Добавление суммы подходящей пары в список
        ans.append(x + y)
# Вывод количества подходящих пар и максимальной суммы среди них
print(len(ans), max(ans))
```

Таким образом, программа последовательно проверяет каждую пару чисел, сохраняет подходящие суммы и выводит итоговые результаты.

Ответ:

71 17664

Задача № 4 (4656)

В файле [17-4.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до $10\,000$ включительно. Определите количество пар, в которых хотя бы один из двух элементов больше, чем среднее арифметическое всех чисел в файле, а их

сумма делится на 7. В ответе запишите два числа: сначала количество найденных пар, а затем – минимальную сумму элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=0h40m0s

Решение

Прежде всего, необходимо вычислить среднее арифметическое всех чисел в файле. Для этого суммируются все элементы списка и результат делится на общее количество элементов. После нахождения среднего арифметического переходим к проверке условий для каждой пары последовательных элементов. Если хотя бы один из элементов пары больше среднего арифметического, и сумма элементов делится на 7 без остатка, такая пара считается удовлетворяющей условиям задачи. Наконец, требуется вывести количество найденных пар и минимальную сумму среди всех подходящих пар.

```
# Читаем содержимое файла '17-4.txt' и преобразуем каждую строку в целое число
a = [int(x) for x in open('17-4.txt')]
# Находим среднее арифметическое всех чисел в файле
avg = sum(a) / len(a)
# Создаём пустой список для хранения сумм подходящих пар
ans = []
# Перебираем пары соседних элементов
for x, y in zip(a, a[1:]):
    # Проверяем, удовлетворяет ли пара условию: хотя бы один элемент больше среднего
    # арифметического,
    # и сумма элементов делится на 7 без остатка
    if (x > avg or y > avg) and (x + y) % 7 == 0:
        # Сохраняем сумму такой пары в список
        ans.append(x + y)
# Выводим количество подходящих пар и минимальную сумму среди них
print(len(ans), min(ans))
```

Ответ:

202 6916

Задача №5 (4688)

В файле [17-243.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000 включительно. Определите количество пар чисел, в которых хотя бы один из двух элементов больше, чем наибольшее из всех чисел в файле, делящихся на 19. В ответе запишите два числа: сначала количество найденных пар, а затем – минимальную сумму элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=0h46m25s

Решение

Первым шагом следует открыть файл и прочитать данные. Далее необходимо найти наибольшее число, которое делится на 19. Для этого создадим новый список, содержащий только те элементы исходного списка, которые делятся на 19 без остатка. Из этого нового списка найдем максимальное значение.

Затем начнем перебирать пары чисел, проверяя каждое число на выполнение условия: хотя бы одно из чисел должно быть больше максимального числа, делящегося на 19. Если оба числа удовлетворяют этому условию, сохраняем их сумму в список. После завершения перебора выводим количество таких пар и минимальную сумму.

```
# Чтение содержимого файла '17-243.txt' и преобразование его в список целых чисел
a = [int(x) for x in open('17-243.txt')]

# Нахождение наибольшего числа, делящегося на 19
m = max([x for x in a if x % 19 == 0])

# Инициализация пустого списка для хранения сумм подходящих пар
ans = []

# Перебор всех пар чисел в списке a
for i in range(len(a) - 1):
    # Проверка условия: хотя бы одно из чисел больше m
    if a[i] > m or a[i + 1] > m:
        # Добавляем сумму пары в список ans
        ans.append(a[i] + a[i + 1])

# Вывод количества пар и минимальной суммы
print(len(ans), min(ans))
```

Ответ:

34 11169

Задание №6 (4705)

В файле [17-257.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые положительные значения до 10 000 включительно. Необходимо найти сумму минимального и максимального числа среди оканчивающихся на 4 элементов последовательности. Затем программа должна найти и вывести количество пар элементов последовательности, сумма которых меньше найденной суммы, а также наибольшую из этих сумм. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=0h56m20s

Решение

Для решения задачи сначала нужно прочитать содержимое файла и преобразовать его в список целых чисел. Затем из этого списка выбираются только те числа, которые оканчиваются на 4, используя операцию взятия остатка при делении на 10. Среди выбранных чисел находятся минимальное и максимальное, после чего вычисляется их сумма. Далее производится проход по

списку чисел, где проверяются каждая пара последовательных элементов. Если сумма такой пары меньше, чем сумма минимального и максимального чисел, оканчивающихся на 4, она сохраняется в специальный список. После проверки всех пар считается количество сохраненных сумм и находится максимальная из них. В итоге выводятся количество пар, сумма которых меньше суммы минимального и максимального чисел, оканчивающихся на 4, и максимальная из этих сумм.

```
a = [int(x) for x in open('17-257.txt')] # Читаем файл и преобразуем данные в список чисел

# Находим минимальное и максимальное число, оканчивающееся на 4
mn = min([x for x in a if x % 10 == 4])
mx = max([x for x in a if x % 10 == 4])

# Вычисляем сумму минимального и максимального числа, оканчивающегося на 4
sum_min_max = mn + mx

# Создаем пустой список для хранения сумм пар
ans = []

# Перебираем пары последовательных элементов
for x, y in zip(a, a[1:]):
    if x + y < sum_min_max: # Если сумма пары меньше суммы минимального и
        # максимального числа
        ans.append(x + y) # Добавляем эту сумму в список ответов

# Выводим количество пар и максимальную из таких сумм
print(len(ans), max(ans))
```

Ответ:

503 10094

Задание № 7(5055)

(Досрочный ЕГЭ-2022) В файле [17-282.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от 0 до 10 000. Определите количество пар элементов последовательности, в которых хотя бы одно число кратно минимальному числу в последовательности, кратному 17. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=1h2m15s

Решение

Для начала необходимо определить минимальное число в последовательности, которое кратно 17. Это делается следующим образом: выбираем все числа из списка, которые делятся на 17 без остатка, и среди них находим наименьшее. Это значение обозначим как m .

Далее нам нужно подсчитать количество пар элементов последовательности, где хотя бы одно число делится на m . Под парой понимается два идущих подряд элемента последовательности. Если хотя бы одно из двух соседних чисел делится на m , то такая пара удовлетворяет условию задачи.

После этого для каждой такой пары мы вычислим сумму её элементов и сохраним эти суммы в отдельный список. Затем найдём максимальное значение среди этих сумм.

Код, который реализует данный алгоритм:

```
# Чтение данных из файла
a = [int(x) for x in open('17-282.txt')]

# Нахождение минимального числа, кратного 17
m = min([x for x in a if x % 17 == 0])

# Список для хранения сумм подходящих пар
ans = []

# Проходим по списку, формируя пары и проверяя условие
for x, y in zip(a, a[1:]):
    # Проверяем, делится ли хотя бы одно из чисел на m
    if x % m == 0 or y % m == 0:
        # Добавляем сумму этой пары к результату
        ans.append(x + y)

# Вывод результата: количество пар и максимальная сумма
print(len(ans), max(ans))
```

Таким образом, сначала мы определяем минимальное число, кратное 17, затем проходим по списку, формируя пары и проверяя каждую пару на выполнение условия, после чего выводим количество таких пар и максимальную сумму их элементов.

Ответ:

24 17613

Задание №8 (5373)

В файле [17-339.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-100\,000$ до $100\,000$ включительно. Определите количество пар последовательности, в которых сумма элементов меньше минимального положительного элемента последовательности, кратного 19. Гарантируется, что такой элемент в последовательности есть. В ответе запишите количество найденных пар, затем абсолютное значение максимальной из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=1h10m0s

Решение

Для начала найдем минимальный положительный элемент последовательности, который делится на 19 без остатка. После этого будем проверять все пары последовательных элементов и считать те из них, у которых сумма меньше найденного минимального кратного 19.

Мы должны выбрать элементы последовательности, которые удовлетворяют двум условиям:

1. Они строго больше нуля ($x > 0$).
2. Они делятся на 19 без остатка ($x \% 19 == 0$).

После фильтрации по этим критериям находим минимальный элемент.

Далее используем функцию `zip`, чтобы пройтись по всем парам последовательных элементов. Для каждой пары проверяем, выполняется ли условие, что их сумма меньше минимального положительного кратного 19. Если условие выполнено, добавляем эту сумму к списку подходящих пар.

Затем определяем максимальную сумму из списка подходящих пар и выводим её абсолютное значение. Важно понимать, что «абсолютная величина» означает, что результат может быть как положительным, так и отрицательным числом, но при выводе оно всегда будет представлено как положительное.

```
# Чтение данных из файла
a = [int(x) for x in open('17-339.txt')]

# Нахождение минимального положительного кратного 19
m = min([x for x in a if x > 0 and x % 19 == 0])

# Список для хранения подходящих сумм
ans = []

# Перебор пар последовательных элементов
for x, y in zip(a, a[1:]):
    # Проверка условия
    if x + y < m:
        ans.append(x + y)

# Вывод результата
print(len(ans), abs(max(ans)))
```

Ответ:

4984 696

Задание №9(6757)

(ЕГЭ-2023) В файле [17-379.txt](#) содержится последовательность натуральных чисел. Элементы последовательности могут принимать целые значения от 1 до 100 000 включительно. Определите количество троек последовательности, в которых только одно из чисел является четырёхзначным, а сумма элементов тройки не меньше максимального элемента последовательности, оканчивающегося на 15. В ответе запишите количество найденных троек, затем максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=1h18m55s

Решение

Сначала определим максимальное число в последовательности, которое заканчивается на 15. Для этого воспользуемся операцией взятия остатка при делении на 100 ($x \% 100$), которая даст нам последние две цифры числа. Нам нужны те числа, у которых остаток равен 15.

Проверка количества четырёхзначных чисел в тройках:

Число считается четырёхзначным, если оно находится в диапазоне от 1000 до 9999. Чтобы проверить, что ровно одно из трёх чисел является четырёхзначным, применим приём подсчёта количества истинных условий с использованием сложения булевых значений. Каждому числу соответствует булево значение (True/False), которое преобразуется в 1 или 0. Важно отметить, что

каждое такое условие должно быть заключено в скобки перед сложением, чтобы обеспечить правильную обработку приоритетов операций. Таким образом, сумма булевых значений покажет, сколько чисел в тройке удовлетворяют условию. Нам нужно, чтобы эта сумма была равна 1. Сумма троек:
Проверим, что сумма трёх чисел больше или равна максимальному числу, заканчивающемуся на 15. Это позволит выбрать подходящие тройки.

```
# Считываем данные из файла
a = [int(x) for x in open('17-379.txt')]

# Находим максимальное число, оканчивающееся на 15
m = max([x for x in a if x % 100 == 15])

# Список для хранения подходящих сумм троек
ans = []

# Перебираем тройки чисел
for x, y, z in zip(a, a[1:], a[2:]):
    # Условие, что ровно одно число является четырёхзначным
    if ((1000 <= x < 10000) + (1000 <= y < 10000) + (1000 <= z < 10000)) == 1 \
        and x + y + z >= m:
        # Добавляем сумму в список
        ans.append(x + y + z)

# Выводим результат: количество троек и максимальная сумма
print(len(ans), max(ans))
```

Ответ:

299 196183

Задание №10 (7226 ред.)

В файле [17-390.txt](#) содержится последовательность целых чисел. Элементы последовательности могут принимать целые значения от $-100\,000$ до $100\,000$ включительно. Определите количество троек, для которых выполняются следующие условия: – ровно два числа в тройке четырёхзначные; – хотя бы одно число в тройке делится на 7; – сумма элементов тройки больше максимального элемента последовательности, запись которого заканчивается на 15. (Гарантируется, что в последовательности есть хотя бы один элемент, запись которого заканчивается на 15.) В ответе запишите количество найденных троек, затем – максимальную из сумм элементов таких троек. В данной задаче под тройкой подразумевается три идущих подряд элемента последовательности.

Ссылка на видео-разбор с таймингом: https://vk.com/video-205546952_456241309?t=1h32m5s

Решение

Для решения поставленной задачи необходимо пройти несколько этапов. Во-первых, нам нужно определить максимальное число в последовательности, которое оканчивается на 15. Это важно, поскольку именно с этим числом будет сравниваться сумма элементов каждой тройки. Мы ищем

такое число, используя оператор взятия по модулю (%), чтобы убедиться, что последние две цифры числа равны 15.

Далее переходим к анализу троек чисел. Под тройкой понимается три последовательных элемента массива. Чтобы перебрать такие тройки, удобно использовать встроенную функцию Python zip. Она позволяет легко получить доступ к трем соседним элементам за одну итерацию.

Теперь проверяем каждое из трех условий для каждой тройки:

1. Условие о количестве четырехзначных чисел: Здесь важно, чтобы ровно два числа были четырехзначными. Четырехзначность определяется диапазоном значений от 1000 до 9999. Поэтому мы проверяем каждый элемент тройки на соответствие этому диапазону и считаем количество таких чисел. Если их ровно два, условие выполнено.
2. Условие делимости на 7: Хотя бы одно число в тройке должно делиться на 7. Это также легко проверить с помощью оператора взятия остатка (%). Если остаток от деления любого из чисел на 7 равен нулю, условие выполняется.
3. Условие сравнения суммы с максимальным элементом: Сумма элементов тройки должна превышать максимальное число, оканчивающееся на 15. Эту проверку мы проводим после того, как убедились, что первые два условия соблюдены.

Если все три условия выполняются, мы сохраняем сумму текущей тройки в специальный список. По завершении перебора всех троек, мы можем вывести количество таких троек и максимальную из их сумм.

Таким образом, решение включает поиск максимального числа, заканчивающегося на 15, перебор троек с проверкой всех условий и сохранение результатов для последующего анализа.

```
a = [int(x) for x in open('файлы к вебу/17-390.txt')]

# Находим максимальное число, оканчивающееся на 15
m = max([x for x in a if abs(x) % 100 == 15])

# Список для хранения сумм подходящих троек
ans = []

# Перебираем тройки чисел
for x, y, z in zip(a, a[1:], a[2:]):
    # Проверяем условия
    if (1000 <= abs(x) < 10000) + (1000 <= abs(y) < 10000) + (1000 <= abs(z) < 10000)
    == 2 and \
        (x % 7 == 0 or y % 7 == 0 or z % 7 == 0) and x + y + z > m:
        # Добавляем сумму тройки в список
        ans.append(x + y + z)

# Вывод результата
print(len(ans), max(ans))
```

Ответ :

22 112649

Telegram: @fast_ege