Задача сложного уровня. З 21

В файле числа от 0 до 100000. Рассматриваются рандомные пары чисел (то есть числа не обязательно должны идти друг за другом).

Необходимо сначала найти количество таких пар, в которых оба числа различны, входят в группу чисел Фибоначчи и при этом числа идут друг за другом в этом самом ряду. Затем минимальную сумму таких пар.

Пример: возьмём первые 10 чисел в этом ряду. 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55. Числа 13 и 21 удовлетворяют условию задачи (они идут подряд), а 21 и 55 нет, как и 3 и 8 (разница в индексах внутри последовательности должна равняться 1).

Решение:1) Заводим список чисел Фибоначчи до 150000 fib.

- 2) Заводим цикл в цикле for (для перебора рандомных пар).
- 3) В список para будем добавлять числа из пары и сразу сортируем в порядке не убывания (clear() нужен для того чтобы очищать числа из предыдущей пары).
- 4) Делаем проверку на то, что числа различны в паре.
- 5) Делаем проверку на то, что оба числа есть в ряду чисел Фибоначчи.
- 6) Смотрим индексы чисел в паре относительно ряда.
- 7) Делаем проверку на то, что сила идут друг за другом.
- 8) Если все проверки пройдены то увеличиваем счётчик, сравниваем с минимальной суммой других пар (берём минимальную).
- 9) Выводим ответ.

```
a = [int(i) for i in open('17hard.txt')]
k = 0
mn = 10 ** 10
para = []
fib = [0, 1]
while fib[-1] <= 150 000:
    teckuchee = fib[-1] + fib[-2]
    fib.append(teckuchee)
for i in range(len(a)):
    for j in range(i+1, len(a)):
        para.clear()
        para.append(a[i])
        para.append(a[j])
        para.sort()
        if a[i] != a[j]:
            if para[0] in fib and para[1] in fib:
                tr1 = fib.index(para[0])
                tr2 = fib.index(para[1])
                if (tr1 + 1) == tr2:
                    k += 1
                    mn = min(mn, a[i] + a[j])
print(k, mn)
```

Задача среднего уровня. 678 9300

В файле числа от 0 до 100 включительно. Рассматриваются рандомные четвёрки. Необходимо посчитать сколько таких четвёрок, где числа могут являться сторонами прямоугольника (квадрат это прямоугольник).

Затем максимальную площадь из всех таких прямоугольников.

## Решение:

- Заводим 4 цикла for внутри друг друга, чтобы перебирались рандомные четвёрки.
- 2) ОБМАН: стороны прямоугольника не могут равняться 0, поэтому делаем проверку на то, чтобы все числа в четвёрке не равнялись 0.
- 3) Заводим список storony, состоящий из чисел из четвёрки, сортируем в порядке неубывания.
- 4) Делаем такую проверку: if storony[0] == storony[1] and storony[2] == storony[3]. Она учитывает и квадрат.
- 5) Если обе проверки пройдены, увеличваем счётчик и считаем максимальную площадь так:  $s = \max(s, storony[0] * storony[3])$ .
- 6) Выводим ответ.

Задача легкого уровня. 6 1887

Камера на дороге общего пользования фиксирует числовое значение номера автомобилей, которые превысили скорость, и сама раздаёт штрафы. Условимся что есть блатные номера. Это трёхзначные номера, состоящие из 3 одинаковых цифр. Например: 777.

Номера 77 и 7777 уже не блатные. Причём штраф получают водители такой, чему рано числовое значение их номера. Товарищ из ГАИ зачем-то решает изучить списки нарушивших. Его особенно привлекают 2 идущих подряд нарушителя в списке на блатных номерах.

Нужно помочь найти количество таких пар, затем сумму штрафа такой пары (просто сложить числа в паре).

## Решение:

- 1) Заводим список блатных номеров alf.
- 2) Заводим цикл for.

- 3) Внутри цикла делаем проверку на то, что a[i] in alf and a[i+1] in alf (оба номера блатные).
- 4) Если проверка пройдена, то увеличиваем счётчик, считаем сумму штрафа и сравниваем с максимальной.
- 5) Выводим ответ.

```
a = [int(i) for i in open('17easy.txt')]
k = 0
mx = 0
alf = [111, 222, 333, 444, 555, 666, 777, 888, 999]
for i in range(len(a)-1):
    if a[i] in alf and a[i+1] in alf:
        k += 1
        mx = max(mx, a[i] + a[i+1])
print(k, mx)
```