

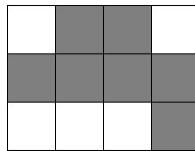
Окончание приема решений 21.07.2024 в 23:59

### Постановка задачи

Исследователь Датум работает с бинарными матрицами. Будем их для красоты изображать в виде пиксельной матрицы (закрашенный пиксель соответствует 1, незакрашенный – 0). Например, матрице

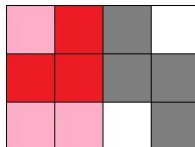
0110  
1111  
0001

соответствует картинка



Датум придумал называть закономерностью такой непустой набор столбцов и соответствующих им значений, что в подматрице образованной столбцами нет указанных значений, а любой непустой собственный поднабор уже этим свойством не обладает.

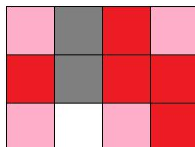
Так, в нашем примере набор столбцов (1,2) со значениями (1,0) является закономерностью.



Действительно, в красной подматрице на рисунке нет строки (1, 0). Но строка (1) есть в подматрице, образованной 1-м столбцом; и строка (0) есть в подматрице, образованной 2-м столбцом. Здесь считаем, что столбцы нумеруются с 1.

Также, набор столбцов (3, 4) со значениями (0, 0) является закономерностью.

Аналогично столбцы (1,3,4) со значениями (0,1,1) являются закономерностью.



В красной подматрице, образованной столбцами с номерами 1, 3, 4 нет строки (0,1,1), зато есть строки вида:

- (1,1,1), т.е. в подматрице со столбцами 3, 4 есть строка (1, 1),

- (0,0,1), т.е. в подматрице со столбцами 1, 4 есть строка (0, 1),
- (0,1,0), т.е. в подматрице со столбцами 1, 3 есть строка (0, 1).

Датум подсчитал, что в указанной матрице ровно 9 закономерностей, в единичной  $n \times n$ -матрице их  $(1 + n(n-1)/2)$ , в матрице  $n \times 1$  есть одна закономерность, если все элементы равны, иначе закономерностей нет, в  $(2^n - 1) \times n$ -матрице с попарно различными строками одна закономерность, в  $2 \times n$ -матрице с нулевой и единичной строкой закономерностей  $n(n-1)$ .

### Задание

Напишите функцию, которая для каждой бинарной матрицы вычисляет число таких закономерностей.

```
import numpy as np

def number_of_patterns(matrix: np.ndarray) -> int:
    # ваш код
    return num

# Создание матрицы
matrix = np.array([[0,0,0,0,0],
                   [1,1,1,0,0],
                   [0,0,1,1,0],
                   [0,0,1,1,1]])

# Получение числа закономерностей
num = number_of_patterns(matrix)

# Вывод результата
print(num)

13
```

Участники должны представить до дедлайна соревнования

файл solution.py, в котором определена функция

```
number_of_patterns(matrix:np.ndarray) -> int
```

Которая возвращает число закономерностей для переданной ей матрицы. Общий алгоритм решения должен быть описан в комментариях в начале программы.

Метрика и способ ее расчета

Ваша функция будет вызвана для некоторого числа матриц, которые участникам не выдаются. При условии совпадения результатов работы вашей функции и правильных ответов, будет зафиксировано время выполнения вызовов функции (в сумме). Это и будет метрика соревнования.

### **Ранжирование участников**

Участники ранжируются по времени выполнения вызовов функции от наименьшего (1 место) к наибольшему (последнее место).

### **Выбор победителей соревнования**

Организаторы выберут некоторое количество участников из топа соревнования, но так же оставляют возможность выбрать участников по оригинальности решения. Победители пройдут на следующий этап отбора (собеседование). Число отобранных участников зависит от общего количества участников, результатов в топе, и т. д. Организаторы не определяют заранее отсечку по метрике.

### **Способ представления решения**

Участники должны воспользоваться телеграм-ботом (см. отдельную инструкцию по боту) для отсылки решения. Отсылать код решения можно неоднократно. При наступлении дедлайна прием решений прекращается.

Каждый раз, когда участник загружает решение с помощью бота, оно ставится в очередь на расчет метрики на приведенном примере матрицы. Это делается, чтобы вы смогли убедиться, что ваше решение представлено в правильном формате и технически с ним нет проблем. Бот пришлет вам подтверждение, что решение запущено успешно или стектрейс ошибки.

### **Требования к решению**

Требуется корректность вычислений для матрицы с числом строк и столбцов до 100. В среде для расчета метрики установлен Python 3.12.4, numpy 2.0.0, scipy 1.14. Кроме стандартных пакетов Python можно использовать только numpy, scipy. Эти пакеты установлены с помощью pip. Другие пакеты нельзя использовать и они не будут доступны на сервере в среде для расчета метрики.

Весь код решения должен быть в одном файле.

Решение должно работать на данной матрице не более 1 минуты на виртуальной машине с 2 ядрами и 4 Гб оперативной памяти.

### **Орг. вопросы**

Участие индивидуальное. Обсуждать варианты решения в группе нельзя. Делиться решениями нельзя. Пользоваться чужими решениями нельзя.

Бейзлайна нет.

Лидерборда во время соревнования нет. Результаты будут опубликованы после подведения итогов.

Учитывается последнее присланное решение.

Если вы нашли баг, пишите @datamove

По работе бота пишите @datamove

**Команда AI Masters желает вам успеха в этом соревновании!**