Информатика. Задания на кластеризацию. Повторение материала.

Фрагмент звёздного неба спроецирован на плоскость с декартовой системой координат. Учёный решил провести кластеризацию полученных точек, являющихся изображениями звёзд, то есть разбить их множество на N непересекающихся непустых подмножеств (кластеров), таких что точки каждого подмножества лежат внутри прямоугольника со сторонами длиной H и W, причём эти прямоугольники между собой не пересекаются. Стороны прямоугольников не обязательно параллельны координатным осям. Гарантируется, что такое разбиение существует и единственно для заданных размеров прямоугольников.

Будем называть центром кластера точку этого кластера, сумма расстояний от которой до всех остальных точек кластера минимальна. Для каждого кластера гарантируется единственность его центра. Расстояние между двумя точками на плоскости $A(x_1, y_1)$ и $B(x_2, y_2)$ вычисляется по формуле:

$$d(A, B) = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)}$$

В файле *А* хранятся координаты точек двух кластеров, где $H = 3$, $W = 3$ для каждого кластера. В каждой строке записана информация о расположении на карте одной точки: сначала координата x, затем координата y. Известно, что количество точек не превышает 1000.

В файле *Б* хранятся координаты точек трёх кластеров, где $H = 3$, $W = 3$ для каждого кластера. Известно, что количество точек не превышает 10 000.

Структура хранения информации в файле *Б* аналогична файлу *А*. Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: P_x – среднее арифметическое абсцисс центров кластеров, и P_y – среднее арифметическое ординат центров кластеров.

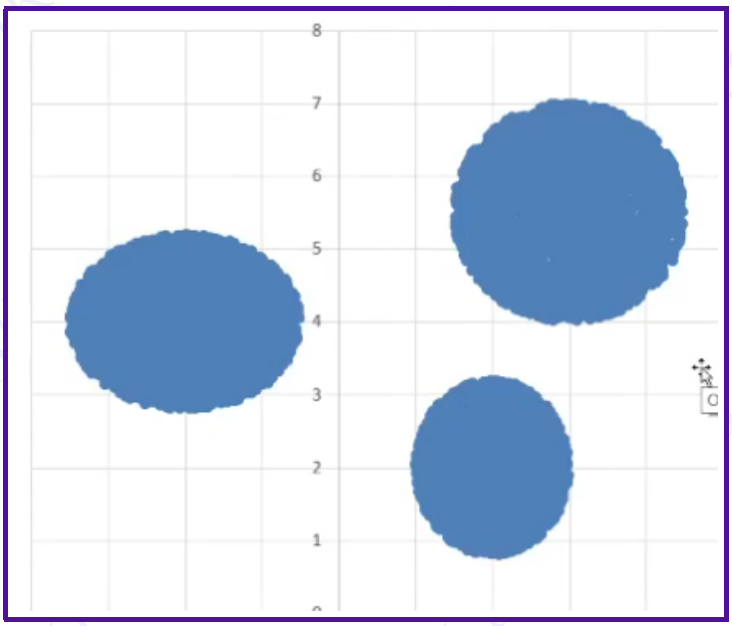
В ответе запишите четыре числа: в первой строке сначала целую часть произведения $P_x \times 10000$, затем целую часть произведения $P_x \times 10000$ для файла *А*, во второй строке – аналогичные данные для файла *Б*.

Разбивка на кластеры и визуализация

На данном вебинаре мы не будем искать центроиды. Наша задача – попробовать новый способ разбивки точек на кластеры, а также визуализировать их для проверки.

Аналогично предыдущим способам, для начала открываем *Excel* и смотрим какую форму имеют кластеры и как они расположены.

В нашем случае они имеют либо форму круга, либо форму овала:



Идея разбивки на кластеры будет следующей: нам нужно найти примерные центры кластеров и их примерный радиус. По этом данным нужно составить уравнение окружности по формуле:

$$(x - x_0)^2 + (y - y_0)^2 = R^2,$$

где (x_0, y_0) – центр кластера, а R – его радиус.

После анализа получаем, что кластеры имеют следующие центры и радиусы (всё приблизительно и в случае ошибки корректируется):

1. $(-2, 4), R = 1.6$
2. $(2, 2), R = 1.3$
3. $(3, 5.5), R = 1.6$

Реализуем такую программу, которая, во-первых, покажет область найденных окружностей. Во-вторых, визуализирует наши кластеры и таким образом станет понятно ошиблись мы с параметрами или разбили точки на кластеры верно.

```
from turtle import *
f = open("27-1b.txt"), s = f.readline()
a = [list(map(float, i.replace(',', '.').split())) for i in f]
m = 100, tracer(0)
pu()
for i in range(-200, 200):  # Циклы для заполнения точками найденных
    for j in range(-200, 200): # уравнений окружности
        x, y = i / 20, j / 20
        if (x + 2)**2 + (y - 4)**2 < 1.6**2: # Рисуем левую окружность
            goto(x*m, y*m)
            dot(3, 'red') # Отображаем окружность красным цветом
        if (x - 2)**2 + (y - 2)**2 < 1.3**2: # Рисуем нижнюю окружность
            goto(x*m, y*m)
            dot(3, 'green') # Отображаем окружность зеленым цветом
        if (x - 3)**2 + (y - 5.5)**2 < 1.6**2: # Рисуем верхнюю окружность
            goto(x*m, y*m)
            dot(3, 'blue') # Отображаем окружность голубым цветом
cl = [[], [], []] # Список для хранения точек для каждого кластера
for i in a: # Проходим по всем точкам в файле
    x, y = i
    if (x + 2)**2 + (y - 4)**2 < 1.6**2: # Если точка попадает в первый круг,
        cl[0].append(i) # добавляем ее в первый кластер
    if (x - 2)**2 + (y - 2)**2 < 1.3**2: # Если попадает во второй круг,
        cl[1].append(i) # добавляем ее во второй кластер
    if (x - 3)**2 + (y - 5.5)**2 < 1.6**2: # Если попадает в третий круг,
        cl[2].append(i) # добавляем ее в третий кластер
for i in cl[0]: # Изменяя cl[0] на cl[1] и cl[2] визуализируем распределение
    x, y = i    # и убеждаемся, что параметры выбраны верно, а так же все
    goto(x*m, y*m) # точки верно распределены по кластерам
    dot(3)
done()
```