

Информатика. Задания на кластеризацию. Функции.

1 Задача

Р-00. (демо-2025) Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, лежащий внутри прямоугольника высотой H и шириной W . Каждая звезда обязательно принадлежит только одному из кластеров.

Истинный центр кластера, или **центроид**, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна. Под расстоянием понимается расстояние Евклида между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ на плоскости, которое вычисляется по формуле:

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

Входные данные

В файле A хранятся данные о звёздах двух кластеров, где $H=3$, $W=3$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y . Значения даны в условных единицах. Известно, что количество звёзд не превышает 1000.

В файле B хранятся данные о звёздах трёх кластеров, где $H=3$, $W=3$ для каждого кластера. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звездах в файле B аналогична файлу A .

Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: Px – среднее арифметическое абсцисс центров кластеров, и Py – среднее арифметическое ординат центров кластеров.

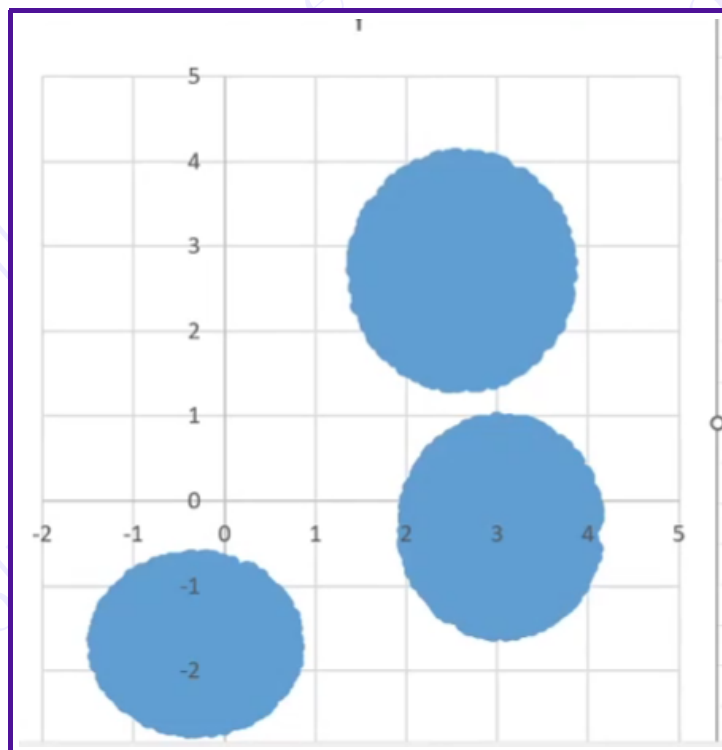
Выходные данные

В ответе запишите четыре числа: в первой строке сначала целую часть произведения $Px \times 10\,000$, затем целую часть произведения $Px \times 10\,000$ для файла A , во второй строке – аналогичные данные для файла B .

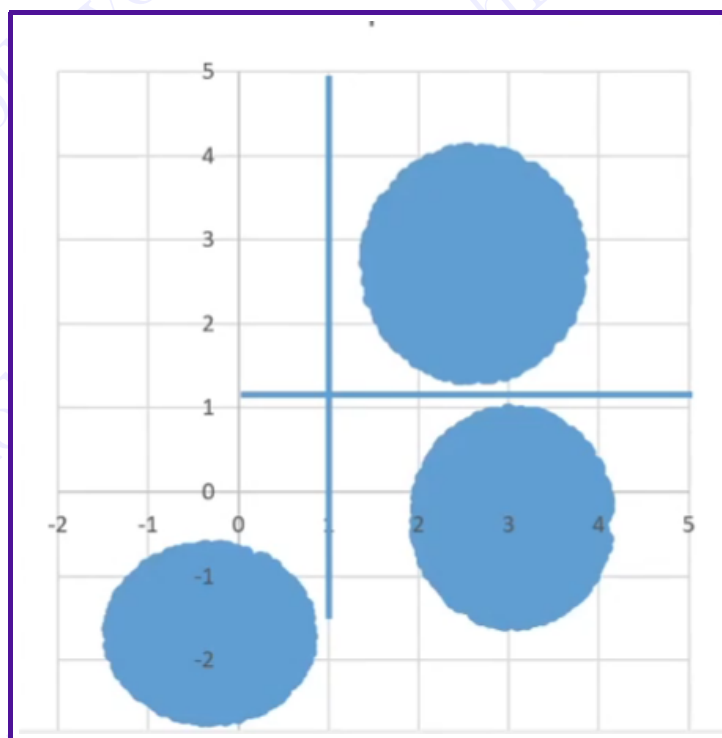
Возможные данные одного из файлов иллюстрированы графиком.

Решение:

Построим диаграмму в Excel:



Первый вариант разделения на кластеры:

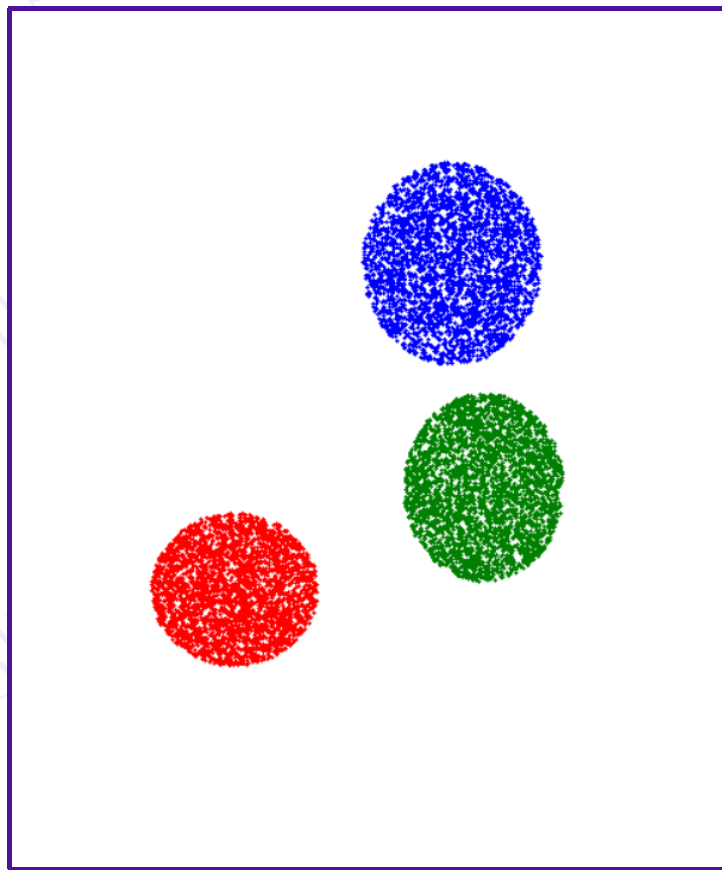


Посмотрим через Черепаху на наши точки внутри одного кластера:

```
from turtle import *

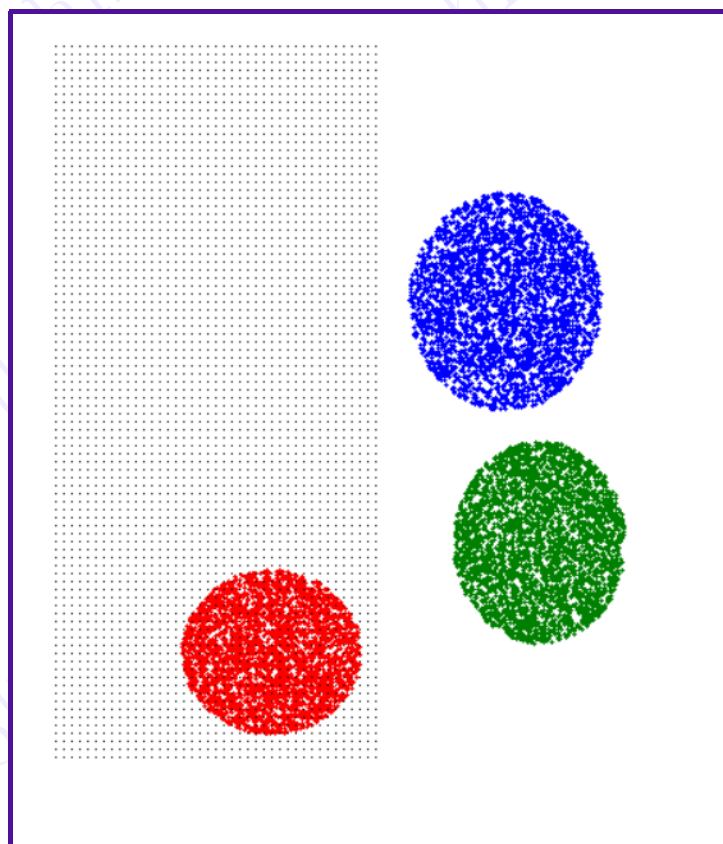
f = open('27-8b.txt') # Открываем файл
s = f.readline() # Считываем данные из файла
a = [list(map(float, i.replace(',', '.').split())) for i in f]
# Заменяем запятую на точку и разделяем по пробелам
cl = [[], [], []] # Кластеры
for i in a: # Проходимся по координатам
    x, y = i # Получаем x и y координаты
    if x < 1: # Первый кластер
        cl[0].append(i) # Добавление в первый кластер
    elif y < 1.2: # Второй кластер
        cl[1].append(i) # Добавление во второй кластер
    else: # Третий кластер
        cl[2].append(i) # Добавление в третий кластер
# Отрисовка кластеров
m = 50
tracer(0)
pu()
color = ['red', 'green', 'blue']
for j in range(3):
    for i in cl[j]:
        x, y = i
        goto(x*m, y*m)
        dot(3, color[j])
done()
```

Тогда получим:

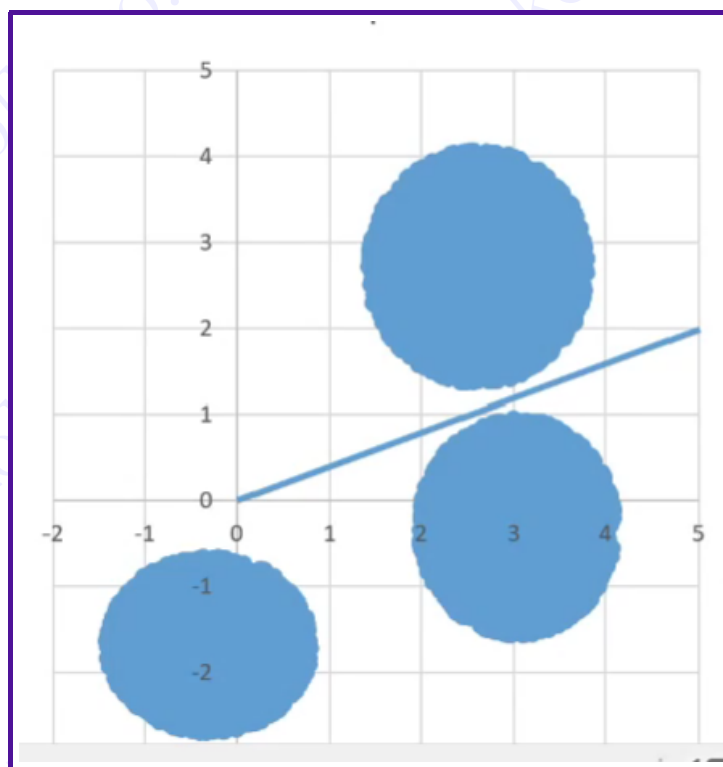


Добавим отрисовку точек:

```
for i in range(-30, 60):  
    for j in range(-30, 60):  
        x, y = i / 10, j / 10  
        if x <= 1:  
            goto(x*m, y*m)  
            dot(3, 'black')  
done()
```



Второе возможное разделение кластеров (по прямой):



Вывод прямой по формуле:

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

$$x_1 = 0, x_2 = 5$$

$$y_1 = 0, y_2 = 2$$

$$\frac{y - 0}{2 - 0} = \frac{x - 0}{5 - 0}$$

$$\frac{y}{2} = \frac{x}{5}$$

Тогда можно заменить условие второго кластера на:

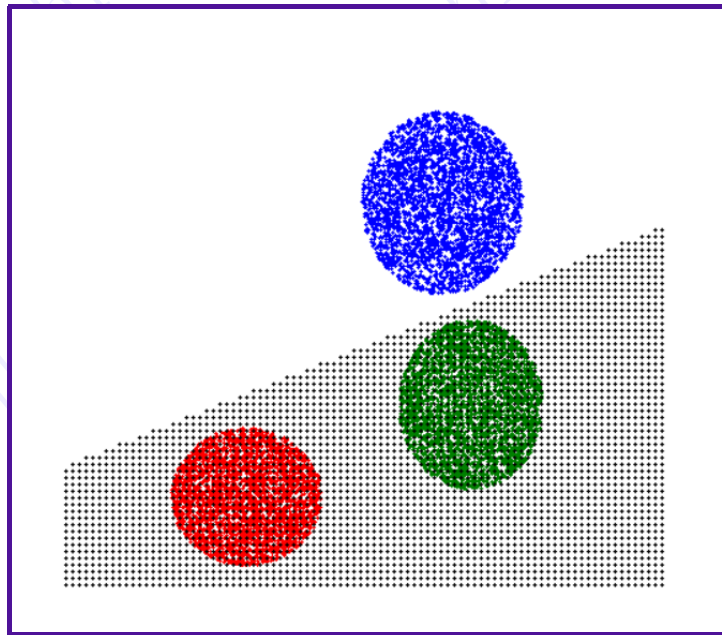
```
elif y / 2 < x / 5: # Второй кластер  
    cl[1].append(i) # Добавление во второй кластер
```

Тогда получим:

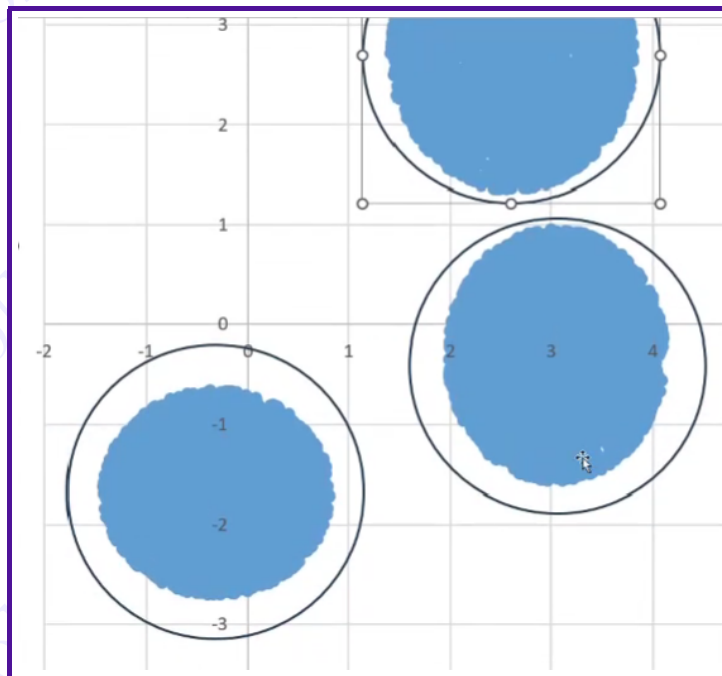
```

from turtle import *
f = open('27-8b.txt') # Открываем файл
s = f.readline() # Считываем данные из файла
a = [list(map(float, i.replace(',', '.').split())) for i in f]
# Заменяем запятую на точку и разделяем по пробелам
cl = [[], [], []] # Кластеры
for i in a: # Проходимся по координатам
    x, y = i # Получаем x и y координаты
    if x < 1: # Первый кластер
        cl[0].append(i) # Добавление в первый кластер
    elif y / 2 < x / 5: # Второй кластер
        cl[1].append(i) # Добавление во второй кластер
    else: # Третий кластер
        cl[2].append(i) # Добавление в третий кластер
# Отрисовка кластеров
m = 50
tracer(0)
pu()
color = ['red', 'green', 'blue']
for j in range(3):
    for i in cl[j]:
        x, y = i
        goto(x*m, y*m)
        dot(3, color[j])
# Отрисовка точек
for i in range(-30, 60):
    for j in range(-30, 60):
        x, y = i / 10, j / 10
        if y / 2 < x / 5:
            goto(x*m, y*m)
            dot(3, 'black')
done()

```



Третий способ разделения по кластерам:



Центр первого c_0 : $y = 1.7, x = -0.3, r = 1.3$

Мы хотим ограничить кластер с помощью окружности, её формула:

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

Тогда получим:

$$(x + 0.3)^2 + (y + 1.7)^2 = 1.3^2$$

Пропишем это и выведем с помощью Черепахи:

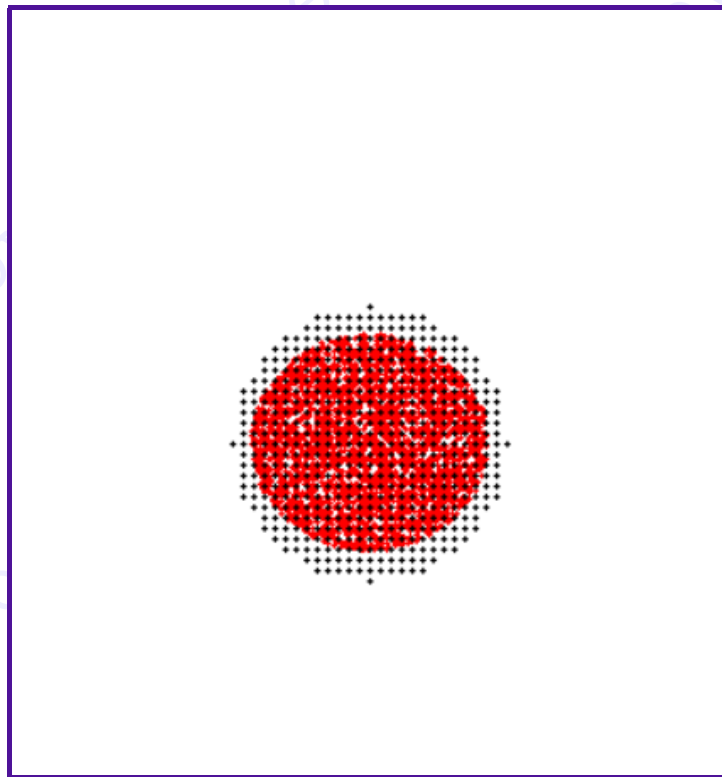
```
from turtle import *

f = open('27-8b.txt') # Открываем файл
s = f.readline() # Считываем данные из файла
a = [list(map(float, i.replace(',', '.').split())) for i in f]
# Заменяем запятую на точку и разделяем по пробелам
cl = [[], [], []] # Кластеры
for i in a: # Проходимся по координатам
    x, y = i # Получаем x и y координаты
    if (x + 0.3) ** 2 + (y + 1.7) ** 2 <= 1.3 ** 2:
        cl[0].append(i)

m = 50
tracer(0)
pu()
color = ['red', 'green', 'blue']
for j in range(3):
    for i in cl[j]:
        x, y = i
        goto(x*m, y*m)
        dot(3, color[j])

for i in range(-30, 60):
    for j in range(-30, 60):
        x, y = i / 10, j / 10
        if (x + 0.3) ** 2 + (y + 1.7) ** 2 <= 1.3 ** 2:
            goto(x*m, y*m)
            dot(3, 'black')
```

done()



Пропишем также для других кластеров:

c1: $y = 2.7$, $x = 2.6$, $r = 1.5$ (берем по большому радиусу)

Тогда уравнение:

$$(x - 2.6)^2 + (y - 2.7)^2 = 1.5^2$$

c2: $y = -0.3$, $x = 3$, $r = 1.4$

Тогда уравнение:

$$(x - 3)^2 + (y + 0.3)^2 = 1.4^2$$

```
from turtle import *
f = open('27-8b.txt') # Открываем файл
s = f.readline() # Считываем данные из файла
a = [list(map(float, i.replace(',', '.').split())) for i in f]
# Заменяем запятую на точку и разделяем по пробелам
```

```

cl = [[], [], []] # Кластеры
for i in a: # Проходимся по координатам
    x, y = i # Получаем x и y координаты
    if (x + 0.3) ** 2 + (y + 1.7) ** 2 <= 1.3 ** 2:
        cl[0].append(i)
    elif (x - 2.6) ** 2 + (y - 2.7) ** 2 <= 1.5 ** 2:
        cl[1].append(i)
    elif (x - 3) ** 2 + (y + 0.3) ** 2 <= 1.4 ** 2:
        cl[2].append(i)

m = 50
tracer(0)
pu()
color = ['red', 'green', 'blue']
for i in range(-60, 120):
    for j in range(-60, 120):
        x, y = i / 20, j / 20
        goto(x * m, y * m)

# Прописываем область для каждого кластера
    if (x + 0.3) ** 2 + (y + 1.7) ** 2 <= 1.3 ** 2:
        dot(3, 'black')
    elif (x - 2.6) ** 2 + (y - 2.7) ** 2 <= 1.5 ** 2:
        dot(3, 'purple')
    elif (x - 3) ** 2 + (y + 0.3) ** 2 <= 1.4 ** 2:
        dot(3, 'orange')

for j in range(3):
    for i in cl[j]:
        x, y = i
        goto(x*m, y*m)
        dot(3, color[j])

done()

```

