

Информатика. Задания на кластеризацию. Повторение материала.

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:

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

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 10000. Структура хранения информации о звездах в файле аналогична файлу B .

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 $P_x \cdot 10000$, затем целую часть произведения $P_y \cdot 10000$ для файла A , во второй строке – аналогичные данные для файла B .

Возможные данные одного из файлов иллюстрированы графиком.

Внимание! График приведён в иллюстративных целях для произвольных значений, не имеющих отношения к заданию. Для выполнения задания используйте данные из прилагаемого файла.

Решение

```
# Импортируем функцию dist из модуля math для вычисления расстояния
from math import dist
f = open('27-1b.txt')
s = f.readline()
a = [list(map(float, i.replace(',', '.').split())) for i in f]
c = [[], [], []] # Список списков точек кластеров
for i in a: # Заполняем кластеры координатами точек
    x, y = i
    if x < 0: # Если абсцисса точки отрицательная,
        c[0].append(i) # то точка принадлежит первому кластеру
    elif y < 3.5: # Иначе если ордината точки меньше 3.5,
        c[1].append(i) # то точка принадлежит второму кластеру
    else: # Иначе точка принадлежит третьему кластеру
        c[2].append(i)
Px = Py = 0 # Среднее арифметическое абсцисс и ординат центров кластеров
for k in range(3): # Проходим по каждому кластеру
    mn = 1000000000500000000 # Минимальное расстояние
    t = [] # Переменная для хранения текущего кандидата на центроид
    for j in range(len(c[k])): # Проходим по каждой звезде в текущем кластере
        star = c[k][j] # Текущая звезда
        sum_l = 0 # Сумма расстояний от текущей звезды до всех остальных
        for i in c[k]: # Вычисляем сумму расстояний до всех звёзд в кластере
            sum_l += dist(star, i) # Добавляем расстояние до звезды i
        if sum_l < mn: # Если сумма расстояний меньше найденного минимального,
            mn = sum_l # то обновляем минимальное расстояние и
            t = star # обновляем кандидата на центроид
    Px += t[0] # Добавляем абсциссу центра к общей сумме
    Py += t[1] # Добавляем ординату центра к общей сумме
print(Px / 3 * 10000, Py / 3 * 10000)
```
