

## Информатика. Задания на кластеризацию. Повторение материала..

### 1 Задача 1

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.

**Тройная звездная система** – система, в которой три звезды находятся на расстоянии не более  $d$  между любыми двумя звездами в системе. Других звезд на расстоянии не более  $d$  быть не должно.

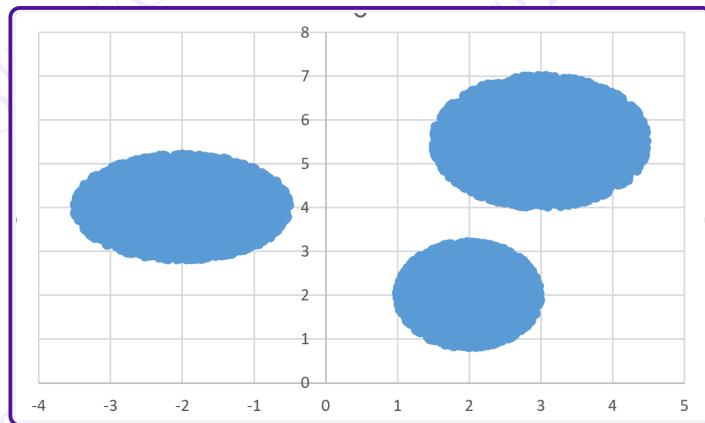
Под расстоянием понимается расстояние Евклида между двумя точками  $A(x_1, y_1)$  и  $B(x_2, y_2)$  на плоскости, которое вычисляется по формуле:

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

Таким образом, три звезды системы представляют из себя вершины треугольника. Необходимо найти треугольник с максимальным периметром в каждом кластере.

#### Решение:

Визуализируем кластеры. Построим на основе данных из файла точечную диаграмму в Excel.



Пусть расстояние  $d = 0.01$

---

```

from math import dist
def dbscan(a, r):
    cl = [] # Массив для кластеров
    while a: # Пока в списке 'a' есть звезды
        cl.append([a.pop(0)]) # Создаем новый кластер
        for i in cl[-1]: # Для каждой звезды в текущем кластере
            for j in a: # Проходим по оставшимся звездам
                if dist(i, j) <= r: # Если рядом есть звезда
                    cl[-1].append(j) # Добавляем звезду в кластер
                    a.remove(j) # Удаляем ее из исходного списка
    return cl # Возвращаем список кластеров

f = open('27-1b.txt')
s = f.readline()
a = [list(map(float, i.replace(',', '.').split())) for i in f]
clusters = dbscan(a, 0.5) # Кластеризация с расстоянием 0.5
d = 0.01 # Максимальное расстояние для тройной звездной системы
for j in clusters: # Для каждого кластера ищем тройную систему
    star_01 = dbscan(j, d) # Кластеризация на звездные системы
    mx = 0 # Максимальный периметр
    mx_star = [] # Координаты звезд с максимальным периметром
    for i in star_01:
        if len(i) == 3: # Если система состоит из трех звезд
            # Вычисляем расстояния между звездами
            d1 = dist(i[0], i[1])
            d2 = dist(i[0], i[2])
            d3 = dist(i[1], i[2])
            if d1 < d and d2 < d and d3 < d: # Если все расстояния меньше d
                if d1+d2+d3 > mx: # Если периметр больше максимального
                    mx = d1+d2+d3 # Обновляем mx
                    mx_star = i # Обновляем координаты звезд
print(mx_star)

```

---

## 2 Задача 2

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.

**Тройная звездная система** – система, в которой три звезды находятся на расстоянии не более  $d$  между любыми двумя звездами в системе. Других звезд на расстоянии не более  $d$  быть не должно.

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:

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

Необходимо найти три звезды в каждом скоплении, которые образуют равносторонний треугольник. Это означает, что расстояния между всеми тремя звёздами должны быть равны, а также равны углы при их соединении.

## Решение:

---

```
from math import dist
def dbscan(a, r):
    cl = [] # Массив для кластеров
    while a: # Пока в списке 'a' есть звезды
        cl.append([a.pop(0)]) # Создаем новый кластер
        for i in cl[-1]: # Для каждой звезды в текущем кластере
            for j in a: # Проходим по оставшимся звездам
                if dist(i, j) <= r: # Если рядом есть звезда
                    cl[-1].append(j) # Добавляем звезду в кластер
                    a.remove(j) # Удаляем ее из исходного списка
    return cl # Возвращаем список кластеров

f = open('27-1b.txt')
s = f.readline()
a = [list(map(float, i.replace(',', ' ').split())) for i in f]
clusters = dbscan(a, 0.5) # Кластеризация с расстоянием 0.5
d = 0.01 # Максимальное расстояние для тройной звездной системы
for j in clusters: # Для каждого кластера ищем тройную систему
    star_01 = dbscan(j, d) # Кластеризация на звездные системы
    mx = 0 # Максимальный периметр
    mx_star = [] # Координаты звезд с максимальным периметром
    for i in star_01:
        if len(i) == 3: # Если система состоит из трех звезд
            # Вычисляем расстояния между звездами
            d1 = dist(i[0], i[1])
            d2 = dist(i[0], i[2])
            d3 = dist(i[1], i[2])
            # По теореме косинусов
            if (d1*d1 < d2*d2 + d3*d3) and (d2*d2 < d1*d1 + d3*d3) and \
                (d3*d3 < d2*d1 + d3*d1):
                print(i) # Печатаем координаты звезд
```

---