

Решение задачи из ЕГЭ №27 «Кластеризация» на PascalABC.Net

Крючков М.В.

V Всероссийская научно-методическая конференция
«Использование системы программирования
PascalABC.Net в обучении программированию»

https://vk.com/ege_maxball
https://t.me/pascalabc_ege



Вариант задачи №27 ЕГЭ

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:

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

В файле A хранятся данные о звёздах двух кластеров, где $H=6$, $W=6$ 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 $H=9$, $W=9$ для каждого кластера. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звёздах в файле B аналогична файлу A.

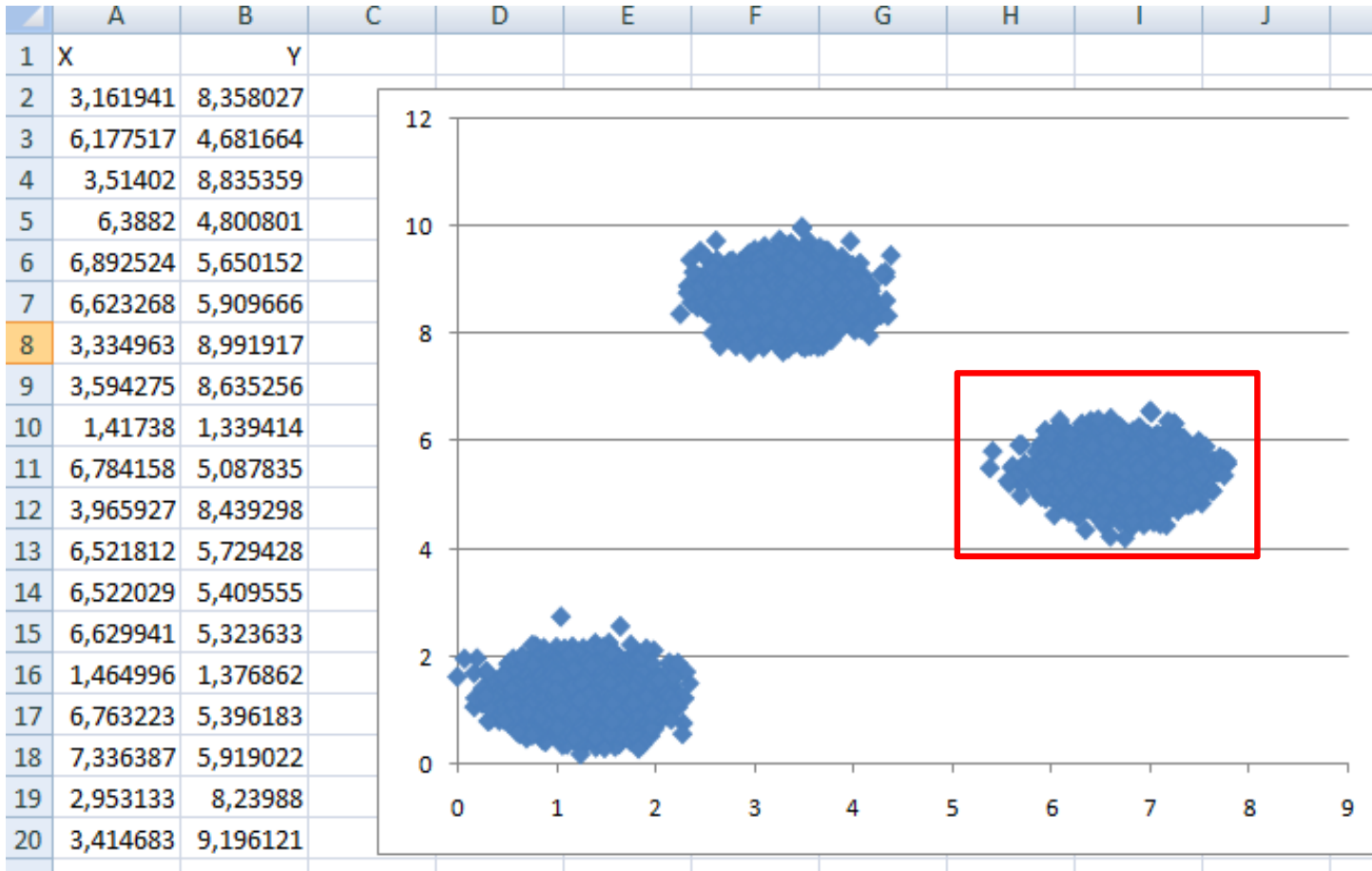
Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: P_x – среднее арифметическое абсцисс центров кластеров, и P_y – среднее арифметическое ординат центров кластеров. В ответе запишите четыре числа: в первой строке сначала абсолютное значение целой части произведения $P_x \times 10000$, затем абсолютное значение целой части произведения $P_y \times 10000$ для файла A, во второй строке – аналогичные данные для файла B.

Рекомендованный способ

1. Загрузить данные в Excel.
2. Построить точечную диаграмму.
3. Проанализировать количество кластеров.
4. Составить формулы разделения по линиям ($y \leq ax + b$) или логические ($y < 10 \ \& \ x < 5$).
5. Составить программу загрузки данных с разделением на кластеры по выведенным формулам.

Минус: не везде есть Excel.

Диаграмма



Разделение

1 кластер:
 $x < 3$ & $y < 4$

2 кластер:
 $x > 5$ & $x < 8$ &
 $y > 4$ & $y < 7$

3 кластер:
 $x > 1$ & $x < 5$ & $y > 6$

Решение на «Python» после анализа в Excel

```
def findClusterNo( x, y ):
    return 0 if x < 3 and y<4 else \
           1 if x >4 and y>4 and y<8 else \
           2
```

```
def dist( p1, p2 ):
    return ((p1[0] - p2[0])**2 + (p1[1] -
    p2[1])**2) ** 0.5
```

```
K = 3
clusters = [ [] for i in range(K) ]
with open('27-b.txt') as F:
    F.readline()
    for s in F:
        x, y = s.replace(',','').split()
        x, y = float(x), float(y)
        clusterNo = findClusterNo( x, y )
        clusters[clusterNo].append( (x, y) )
centers = []
```

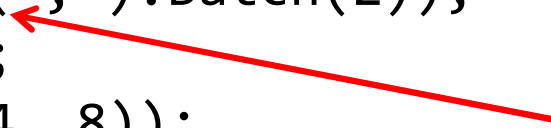
```
for k in range(K):
    minSumDist = float('inf')
    for pCenter in clusters[k]:
        sumDist = sum( dist(pCenter, p)
                        for p in clusters[k] )
        if sumDist < minSumDist:
            minSumDist = sumDist
            center = pCenter
    centers.append( center )

sumX, sumY = 0, 0
for k in range(K):
    sumX += centers[k][0]
    sumY += centers[k][1]
    print(centers[k][0], centers[k][1])
print( sumX/K*10000, sumY/K*10000 )
```

Проблема десятичной точки/запятой

Решение на PascalABC.Net после анализа в Excel

```
##      // без модулей и функций в старой версии языка
var t:=MatrByRow(ReadAllText('27-b.txt').ToReals(',').Batch(2));
var KL0:=t.Rows.Where(\(x,y) -> (x<3) and (y>4));
var KL1:=t.Rows.Where(\(x,y) -> (x>4) and (y in 4..8));
var KL2:=t.Rows.Where(\(x,y) -> (x in 1..5) and (y>6));
```



```
var (kx, ky):=(|0.0|*3, |0.0|*3);
(kx[0], ky[0]):=KL0.OrderBy(\(cx, cy) ->
    KL0.Sum(\(x,y) -> sqrt((cx-x)**2 + (cy-y)**2))).First;
(kx[1], ky[1]):=KL1.OrderBy(\(cx, cy) ->
    KL1.Sum(\(x,y) -> sqrt((cx-x)**2 + (cy-y)**2))).First;
(kx[2], ky[2]):=KL2.OrderBy(\(cx, cy) ->
    KL2.Sum(\(x,y) -> sqrt((cx-x)**2 + (cy-y)**2))).First;
Println(kx.Average*10000, ky.Average*10000);
```

Второй вариант после анализа в Excel

```
## uses coords;
// функция определения номера кластера
function NumCl(p:Point):= (p.x<3) and (p.y>4) ? 0 :
                        (p.x>4) and (p.y in 4..8) ? 1 : 2;
// считать из файла в список точек (Point)
var t:=ReadAllText('27b1.txt').ToReals(',').Batch(2, \ (x,y) -> Pnt(x,y));
var SummaP:=Pnt(0,0); // сумма координат центров
for var n:=0 to 2 do begin
    var CL:=t.Where(p -> NumCl(p)=n).ToArray; // список звезд кластера
    // определение центроида (медоида)
    var center:=CL.OrderBy(p -> CL.Sum(q -> Distance(p, q))).First;
    SummaP.X += center.x;
    SummaP.Y += center.y;
end;
Println(SummaP.x*10000/3, SummaP.y*10000/3);
```

Программное построение точек для анализа границ кластеров

Python

```
from turtle import *
tracer(0)
hideturtle()
screensize(2000, 2000)
scale=50
up()
setpos(-1000,0)
down()
goto(1000,0)
up()
setpos(0,-1000)
down()
goto(0,1000)
data = []
up()
with open('27-b.txt') as F:
    F.readline()
    for s in F:
        x, y = s.replace(',','').split()
        x, y = float(x), float(y)
        data.append( (x, y) )
        goto( x*scale, y*scale )
        dot( 3, 'black' )
update()
done()
```

Работает медленно, масштабировать перезапуском,
есть шанс упустить аномалии, оси без чисел.

PascalABC.Net

```
## uses coords;
var t:=ReadAllText('27-b.txt').ToReals(',')
    .Batch(2, \ (x,y) -> Pnt(x,y));
DrawPoints(t.ToArray, 3);
```

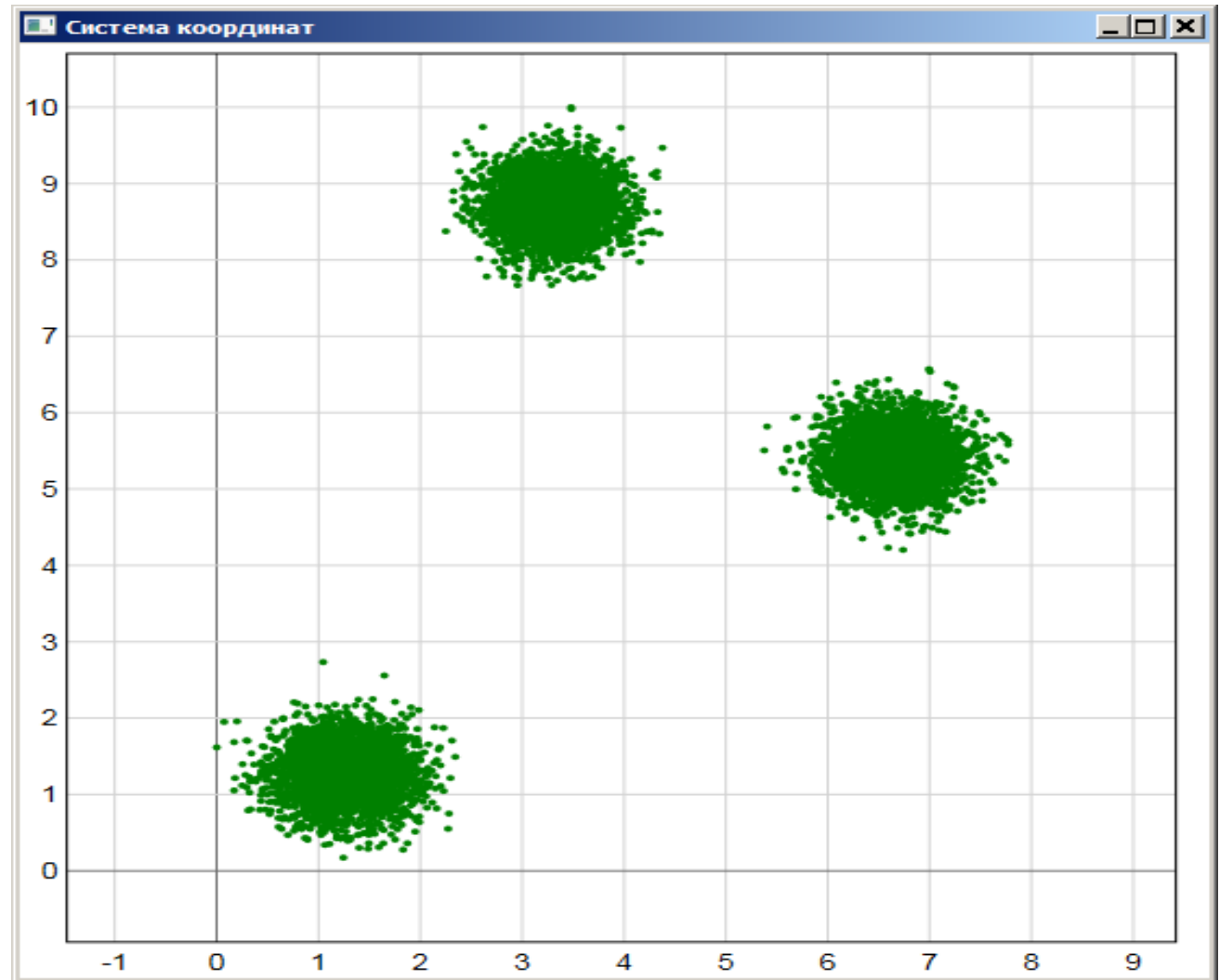
Работает мгновенно, можно масштабировать и
двигать мышкой.

Выводы:

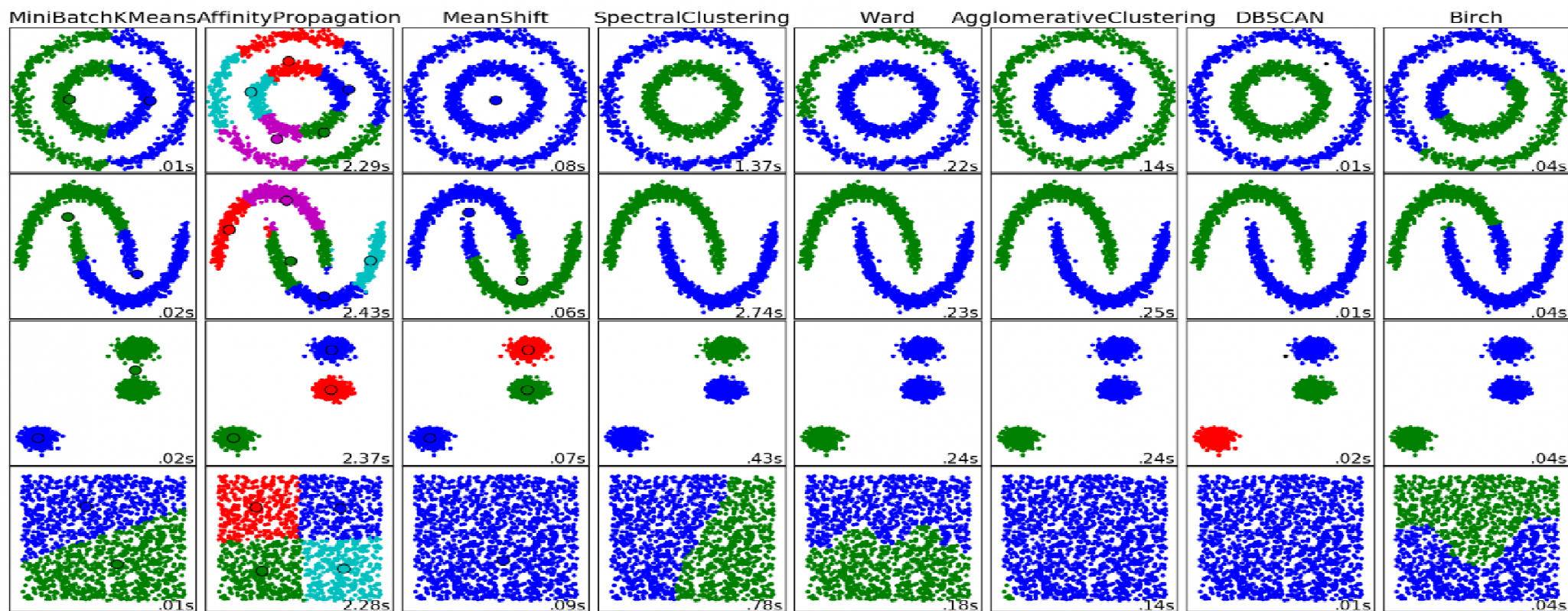
1. Построение точек в Python хуже, чем в Excel.
2. На PascalABC.Net можно сразу решать, не теряя время на Excel.

Окно вывода точек

Кластеры легко
разделить



Варианты алгоритмов автоматического разбиения кластеров

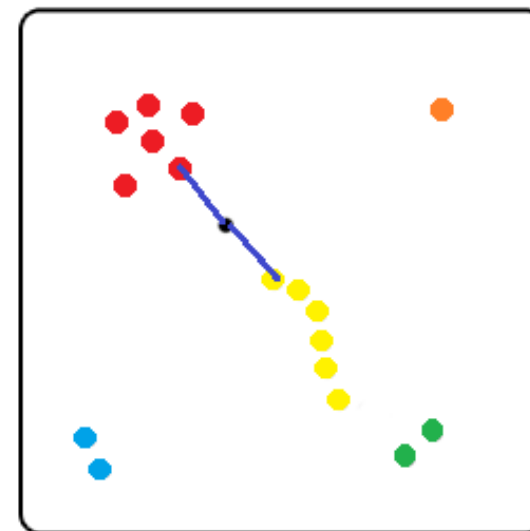
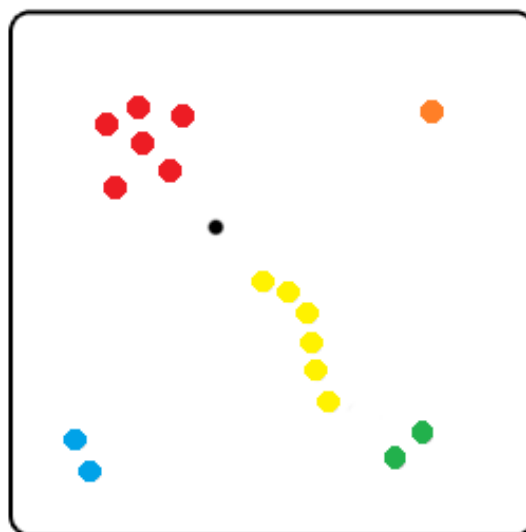
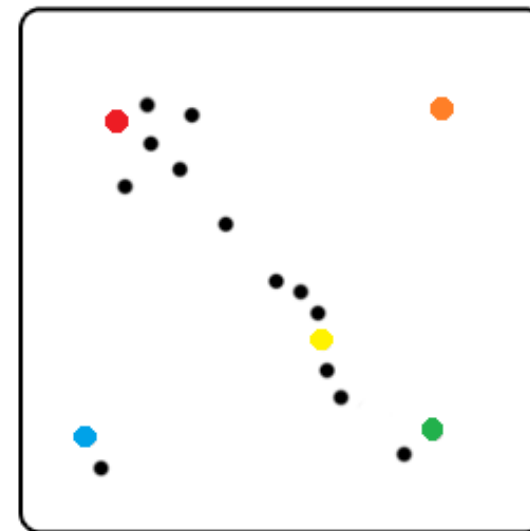
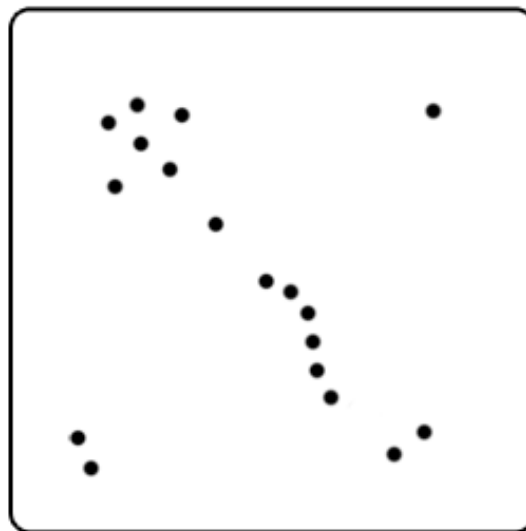


Агломеративный алгоритм или метод «ближнего соседа»

Ввести понятия: **Epsilon** - максимальное расстояние от любой точки до ближайшей в кластере; **Аномалия** – кластер с малым количеством звезд.

1. Взять одну точку – она принадлежит кластеру №0.
2. Взять следующую точку. Выявить список кластеров, которым точка может принадлежать. Расстояние от нее до какой-либо (Any) точки в кластере меньше Epsilon.
 - a) Если таких кластеров нет, то создать новый кластер из одной точки.
 - b) Если «близкий» кластер один, то точку присоединить к нему.
 - c) Если несколько кластеров (точка их соединяет), то:
 - i. создать новый кластер, включающий в себя все «близкие» кластеры и новую точку.
 - ii. удалить кластеры, которые стали частью нового большого
3. Искать центроиды в каждом кластере предыдущим способом.

Этапы создания кластеров



Код алгоритма

```
## uses coords;
var t:=ReadAllText('27-b.txt').ToReals(',').Batch(2, \((x,y) -> Pnt(x,y));
var CL:=lst(lst(t.First));           // список списков звёзд
foreach var p in t.Skip(1) do begin
    // определить номера «близких» кластеров
    var numCL:= CL.Indices(c -> c.Any(g -> Distance(g, p) < 1)).ToArray;
    // если нет таких, то создать новый кластер
    if numCL.Count=0 then CL.Add(lst(p))
    else begin                       // иначе добавить точку в первый попавшийся
        CL[ numCL[0] ].Add(p);
        if numCL.Count > 1 then begin
            // если кластеров больше одного, то сделать новый кластер склейкой этих
            CL.Add(lst( numCL.SelectMany(n -> cl[n]) ));
            // а потом удалить отдельные прежние
            foreach var k in numCL.Reverse do CL.RemoveAt(k);
        end;
    end;
end;
end;
```



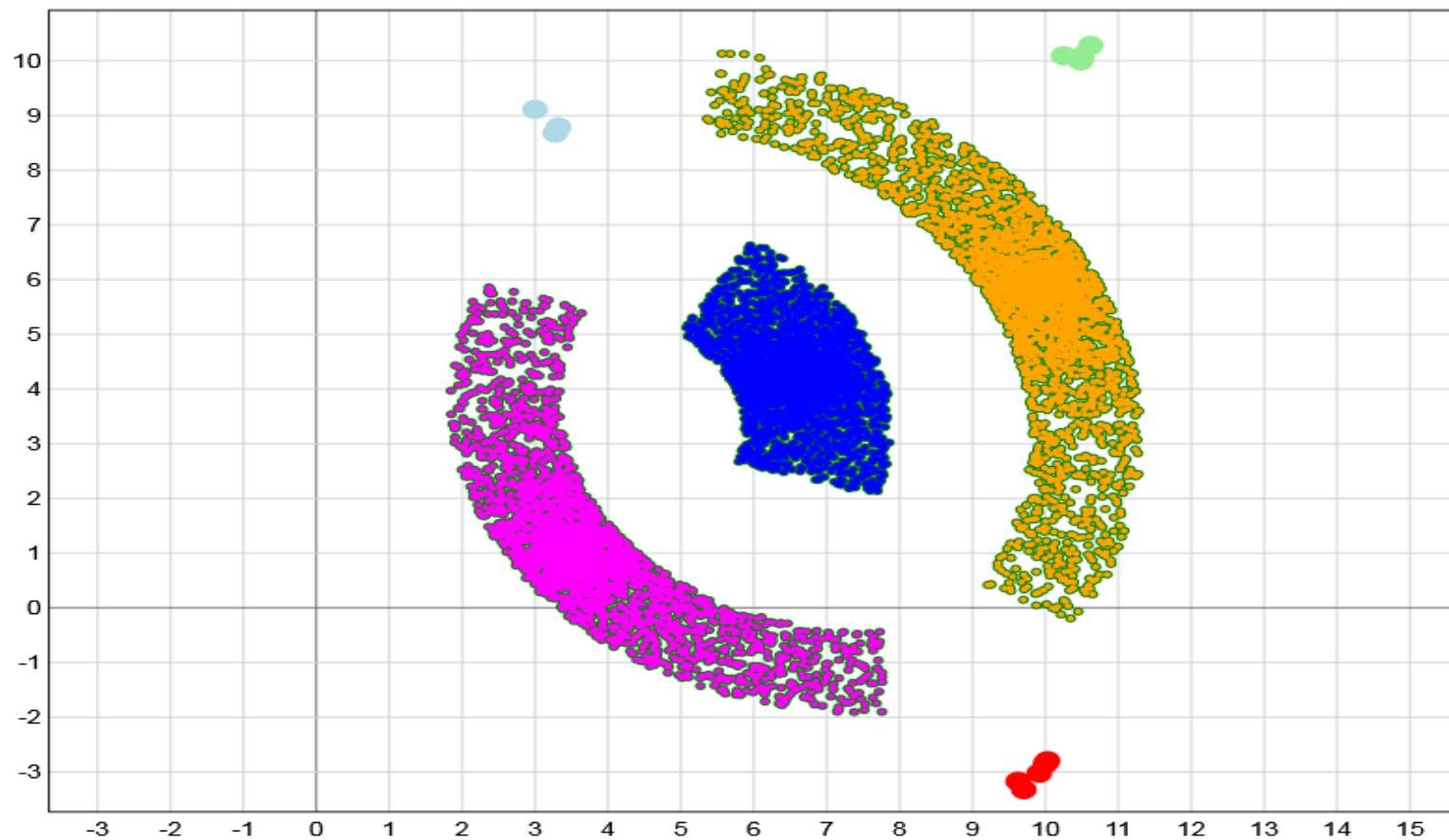
Epsilon

Поиск центров (медоид)

После автоматического разбиения на кластеры поиск центров в цикле можно визуализировать перерисовкой кластеров с выделением аномалий (кластеров с малым количеством точек).

```
var centers:= 1st(|Pnt(0,0)|*0);           // список центров
foreach var c in CL do begin
    var k:=c.count;                       // количество звезд в кластере
    if k>30 then // берем в учет только большие кластеры
        centers.Add( c.OrderBy(p -> c.Sum(q -> Distance(p, q))).First );
    // нарисовать точки с выделением кластеров с малым количеством звезд
    DrawPoints(c.ToArray, k>30 ? 2 : 7);
end;
centers.PrintLines;                       // список центров вывести
// вывод среднего арифметического по условию задачи
print(centers.Average(p-> p.x) * 10000, centers.Average(p-> p.y) * 10000);
```

Вариант вывода кластеров



Полная программа без комментариев

```
## uses coords;
var (Epsilon, MinCount) := (0.3, 30);
var t:= ReadAllText('27-b.txt').ToReals(',').Batch(2, \((x,y) -> Pnt(x,y));
var CL:= lst( lst( t.First ) );
foreach var p in t.Skip(1) do begin
    var numCL:= CL.Indices(c -> c.Any(g -> Distance(g, p) <= Epsilon)).ToArray;
    if numCL.Count=0 then CL.Add(lst(p))
    else begin
        CL[ numCL[0] ].Add(p);
        CL.Add( lst( numCL.SelectMany(n -> CL[n]) ) );
        foreach var k in numCL.OrderDescending do CL.RemoveAt(k);
    end;
end;
var centers:= lst(|Pnt(0,0)|*0); // или new List<Point>;
foreach var c in CL do begin
    if c.Count > MinCount then centers.Add(c.orderby(p -> c.Sum(q -> Distance(p,q))).First);
    DrawPoints(c.ToArray, c.Count > MinCount ? 2 : 7);
end;
centers.PrintLines;
println(centers.Average(p-> p.x)*100000, centers.Average(p-> p.y)*100000);
```

