

27

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:

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

В файле А хранятся координаты точек двух кластеров, где $H = 6,5$ и $W = 4,5$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x , затем координата y . Известно, что количество точек не превышает 1000.

В файле Б хранятся координаты точек трёх кластеров, где $H = 6,5$, $W = 5$ для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации в файле Б аналогична структуре в файле А.

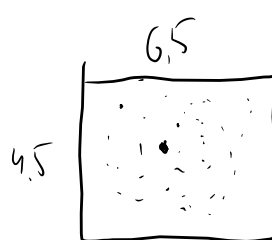
Известно, что в файле А имеются координаты ровно двух, а в файле Б ровно трёх «лишних» точек, представляющих аномалии, которые возникли в результате помех при передаче данных. Эти точки не относятся ни к одному из кластеров, их учитывать не нужно.

Для файла А определите координаты центра каждого кластера, затем найдите два числа: P_1 — минимальное расстояние между центром одного кластера и точкой другого кластера и P_2 — максимальное расстояние между центром кластера и точкой другого кластера.

Для файла Б определите координаты центра каждого кластера, затем найдите два числа: Q_1 — среднее арифметическое расстояний от центра кластера с минимальным количеством точек до точек этого кластера и Q_2 — среднее арифметическое расстояний от центра кластера с максимальным количеством точек до точек этого кластера. Гарантируется, что во всех кластерах количество точек различно. Нулевое расстояние от центра кластера до самого себя не учитывается.

В ответе запишите четыре числа: в первой строке — сначала целую часть произведения $P_1 \times 10\,000$, затем целую часть произведения $P_2 \times 10\,000$; во второй строке — сначала целую часть произведения $Q_1 \times 10\,000$, затем целую часть произведения $Q_2 \times 10\,000$.

Возможные данные одного из файлов проиллюстрированы графиком.



```

1  from math import dist
2
3  clustersA = ([], [])
4  clustersB = ([], [], [])
5
6  with open('01_27var01A.txt') as file:
7      for s in file:
8          s = s.replace(__old: ',', __new: '.')
9          x, y = map(float, s.split())
10         if x > 0 and y > 16:
11             clustersA[0].append((x, y))
12         elif x > 0 and 10 < y < 15:
13             clustersA[1].append((x, y))
14     print(* (sm := tuple(len(kl) for kl in clustersA)), '|', sum(sm))
15
16     with open('01_27var01B.txt') as file:
17         for s in file:
18             s = s.replace(__old: ',', __new: '.')
19             x, y = map(float, s.split())
20             if x > 12 and 13 < y < 19:
21                 clustersB[0].append((x, y))
22             elif 13 < x < 18 and 6 < y < 13:
23                 clustersB[1].append((x, y))
24             elif 23 < x < 29:
25                 clustersB[2].append((x, y))
26     print(* (sm := tuple(len(kl) for kl in clustersB)), '|', sum(sm))
27
28
29 def get_centroid(kl): 2 usages  Сергей Анатольевич
30     res = []
31     for point in kl:
32         sum_dist = sum(dist(point, p) for p in kl)
33         res.append((sum_dist, point))
34     return min(res)[1]
35
36
37 centroidsA = tuple(get_centroid(kl) for kl in clustersA)
38 centroidsB = tuple(get_centroid(kl) for kl in clustersB)
39 print('---' * 5)
40
41
42 def get_P(): 1 usage  Сергей Анатольевич
43     dist_cent_point = [dist(centroidsA[i], p) for i in range(2) for p in clustersA[-1 + i]]
44     return min(dist_cent_point), max(dist_cent_point)
45
46
47 def get_Q(n_kl): 1 usage  Сергей Анатольевич
48     return int(sum(dist(centroidsB[n_kl], p) for p in clustersB[n_kl]) / (len(clustersB[n_kl]) - 1) * 10_000)
49
50
51 print(* (abs(int(x * 10_000)) for x in get_P()))
52 print(* (get_Q(n) for n in (0, 2)))

```


Статград тренировочная работа №1 2025.10.23 В1

27

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

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

В файле А хранятся координаты точек **двух** кластеров, где $H=8$, $W=4$ 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 точек не превышает 1000.

В файле Б хранятся координаты точек **трёх** кластеров, где $H=6$, $W=7$ для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации о звёздах в файле Б аналогична структуре в файле А.

Известно, что в файле Б имеются координаты ровно трёх «лишних» точек, представляющих аномалии, которые возникли в результате помех при передаче данных. Эти три точки не относятся ни к одному из кластеров, их учитывать не нужно.

Для файла А определите координаты антицентра каждого кластера, затем вычислите два числа: P_1 – сумма абсциссы и ординаты антицентра кластера с наименьшим количеством точек, и P_2 – сумма абсциссы и ординаты антицентра кластера с наибольшим количеством точек. Гарантируется, что во всех кластерах количество точек различно.

Для файла Б определите координаты антицентра каждого кластера, затем вычислите два числа: Q_x – абсциссу наиболее отдалённого антицентра кластера от начала координат, и Q_y – ординату ближайшего антицентра кластера к началу координат.