

DS-7. AiСД. 3.

16 янв 2022, 18:48:17

старт: 26 дек 2021, 12:29:39

финиш: 25 янв 2022, 14:29:39

до финиша: 8д. 19ч.

начало: 26 дек 2021, 12:29:39

конец: 25 янв 2022, 14:29:39

длительность: 30д. 2ч.

С. Сортировка точек

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Выведите все исходные точки в порядке возрастания их расстояний от начала координат. Создайте структуру Point и сохраните исходные данные в массиве структур Point.

Формат ввода

Программа получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 .

Формат вывода

Необходимо вывести все исходные точки в порядке возрастания их расстояний от начала координат.

Пример

Ввод

```
2
1 2
2 3
```

Вывод

```
1 2
2 3
```

Язык Python 3.7.3

[Набрать здесь](#)[Отправить файл](#)

```
14 def extract(self):
15     value = self.tree_[1] #замена значения на магнитуду
16     self.tree_[1] = self.tree_[self.size_]
17     self.size_ -= 1
18     p_ex = 1
19     while p_ex * 2 <= self.size_:
20         if p_ex * 2 == self.size_:
21             min_idx = p_ex * 2 #замена названия переменной
22         elif self.tree_[p_ex*2].magnitude < self.tree_[p_ex*2+1].magnitude: #замена знака сравнения и значения на маг
23             min_idx = p_ex*2
24         else:
25             min_idx = p_ex * 2 + 1
26         if self.tree_[p_ex].magnitude > self.tree_[min_idx].magnitude: #замена знака сравнения и значения на магн
27             self.tree_[p_ex], self.tree_[min_idx] = self.tree_[min_idx], self.tree_[p_ex]
28         p_ex = min_idx
29     return value.x, value.y
30
31 class Point:
32     def __init__(self, x_inp, y_inp):
33         self.x = x_inp
34         self.y = y_inp
35         self.magnitude = x_inp ** 2 + y_inp ** 2
36
37 file = open('input.txt')
38 n = int(next(file))
39
40 point_Heap = Heap()
41
42 for i in range(0, n):
43     x, y = tuple(map(int, next(file).split()))
44     point_Heap.add_Point(Point(x, y))
45
46 otvet = ''
47 for i in range(0, n):
48     x, y = point_Heap.extract()
49     otvet += f'{x} {y}\n'
50 print(otvet)
```

ОтправитьПредыдущая