

Тренировки по алгоритмам 5.0 от Яндекса — Занятие 3 (Множества и словари)

2 апр 2024, 06:53:25
старт: 15 мар 2024, 22:30:00
финиш: 27 мар 2024, 20:00:00
длительность: 11д. 21ч.
...

Объявления жюри

Ваше участие в соревновании завершено. Вы можете дорешивать задачи и отправлять решения вне соревнования

Положение участников Задачи Посылки

G. Построить квадрат

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Задано множество, состоящее из N различных точек на плоскости. Координаты всех точек — целые числа. Определите, какое минимальное количество точек нужно добавить во множество, чтобы нашлось четыре точки, лежащие в вершинах квадрата.

Формат ввода

В первой строке вводится число N ($1 \leq N \leq 2000$) — количество точек.

В следующих N строках вводится по два числа x_i, y_i ($-10^8 \leq x_i, y_i \leq 10^8$) — координаты точек.

Формат вывода

В первой строке выведите число K — минимальное количество точек, которые нужно добавить во множество.

В следующих K строках выведите координаты добавленных точек x_i, y_i через пробел.

Координаты должны быть целыми и не превышать 10^9 по модулю.

Если решений несколько — выведите любое из них.

- ✓ A. Плейлисты
- ✓ B. Анаграмма?
- ✓ C. Удаление чисел
- ✓ D. Повторяющееся число
- ✓ E. Два из трех
- ✓ F. Замена слов
- ✓ G. Построить квадрат
- ✓ H. Спички детям не игрушка!
- ✓ I. Играйте в футбол!
- ✓ J. P2P обновление

Пример 1

Ввод	Вывод
2 0 1 1 0	2 0 0 1 1

Пример 2

Ввод	Вывод
3 0 2 2 0 2 2	1 0 0

Пример 3

Ввод	Вывод
4 -1 1 1 1 -1 -1 1 -1	0

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь Отправить файл

```
1 import math
2
3 def get_quads(p00, p01):
4     dx = p01[0]-p00[0]
5     dy = p01[1]-p00[1]
6
7     p10r = (p00[0]+dy, p00[1]-dx)
8     p11r = (p01[0]+dy, p01[1]-dx)
9
10    p10l = (p00[0]-dy, p00[1]+dx)
11    p11l = (p01[0]-dy, p01[1]+dx)
12
13    return [p10r, p11r], [p10l, p11l]
14
15 def answer(points):
16     points_set = set(points)
17
18     max_answer = tuple()
19     for i in range(len(points)):
20         cur_answer = tuple([points[i]])
21
22         for j in range(i+1, len(points)):
23             cur_answers = [[points[i], points[j]], [points[i], points[j]]]
24             quads = get_quads(points[i], points[j])
25
26             for k in range(len(cur_answers)):
27                 for point in quads[k]:
28                     if point in points_set:
29                         cur_answers[k].append(point)
30
31             cur_answer = tuple(cur_answers[0])
32             for k in range(len(cur_answers)):
33                 if len(cur_answers[k]) > len(cur_answer):
34                     cur_answer = tuple(cur_answers[k])
35
36             if len(cur_answer) > len(max_answer):
37                 max_answer = cur_answer
38
```

Отправить

осталось 100 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы	
16 мар 2024, 02:12:57	109810527	G	Python 3.9 (PyPy 7.3.11)	OK	-	0.947s	28.33Mb	-	-	отчёт
16 мар 2024, 02:11:54	109810488	G	Python 3.12.1	TL	-	2.076s	4.14Mb	31	-	отчёт
16 мар 2024, 00:57:32	109806404	G	Python 3.12.1	WA	-	57ms	4.20Mb	4	-	отчёт