

Яндекс. Тренировки по алгоритмам 2.0, занятие 7 (B)

🕒 2 апр 2024, 06:14:40

старт: 23 сен 2021, 14:00:00

...

Объявления жюри

Положение участников

Задачи

Посылки

С. Минимальное покрытие

Ограничение времени	3 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На прямой задано некоторое множество отрезков с целочисленными координатами концов $[L_i, R_i]$. Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок $[0, M]$, (M — натуральное число), содержащее наименьшее число отрезков.

Формат ввода

В первой строке указана константа M ($1 \leq M \leq 5000$). В каждой последующей строке записана пара чисел L_i и R_i ($L_i, R_i \leq 50000$), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 100 000.

Формат вывода

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка $[0; M]$. Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно. Если покрытие отрезка $[0, M]$ исходным множеством отрезков $[L_i, R_i]$ невозможно, то следует вывести единственную фразу "No solution".

Пример 1

Ввод	Вывод
1 -1 0 -5 -3 2 5 0 0	No solution

Пример 2

Ввод	Вывод
1 -1 0 0 1 0 0	1 0 1

Язык

Python 3.12.1

▼

Набрать здесь

Отправить файл

```
1 fin = open('input.txt')
2 M = int(fin.readline())
3 events = []
4
5 TYPE_START = 10
6 TYPE_END_M = 15
7 TYPE_END = 20
8 TYPE_START_M = 25
9
10 segments, id = [], 0
11
12 line = fin.readline().rstrip()
13 while line != '0 0':
14     l, r = [int(x) for x in line.split()]
15     segments.append((l, r))
16     events.append((l, TYPE_START, id))
17     events.append((r, TYPE_END, id))
18     id += 1
19     line = fin.readline().rstrip()
20 events.append((0, TYPE_START_M, -1))
21 events.append((M, TYPE_END_M, -1))
22
23 def answer(events, segments):
24     events.sort()
25     res = []
26
27     curr_started, curr_needed, start_check = set(), set(), False
28     for event in events:
29         _, event_type, id = event
30
31         if event_type == TYPE_START:
32             curr_started.add(id)
33         elif event_type == TYPE_START_M:
34             start_check = True
35             curr_needed = curr_started.copy()
36             if not curr_needed: return 'No solution'
37         elif event_type == TYPE_END_M:
38             start_check = False
```

Отправить

📘 осталось 97 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы	
25 ноя 2023, 04:07:40	99107155	C	Python 3.12.1	OK	-	2.018s	35.23Mb	-	-	отчёт
25 ноя 2023, 04:06:49	99107130	C	Python 3.12.1	OK	-	2.069s	35.23Mb	-	-	отчёт
25 ноя 2023, 04:06:19	99107118	C	Python 3.9 (PyPy 7.3.11)	ML	-	0.991s	95.55Mb	15	-	отчёт