

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

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 <input type="text"/>	Вывод <input type="text"/>
1 -1 0 -5 -3 2 5 0 0	No solution

Пример 2

Ввод <input type="text"/>	Вывод <input type="text"/>
1 -1 0 0 1 0 0	1 0 1

Язык Python 3.12.1

Набрать здесь

Отправить файл

```
1 # считываем данные
2 M = int(input().strip()) # правый край отрезка
3 segments = []
4 start, end = map(int, input().split())
5 while start != 0 or end != 0:
6     if end > 0 and start < M:
7         segments.append((start, end))
8     start, end = map(int, input().split())
9
10 segments.sort()
11 answer = []
12 cur_rg = 0
13 next_rg = 0
14 cur_best = 0, 0
15 for segment in segments:
16     if segment[0] > cur_rg:
17         answer.append(cur_best)
18         cur_rg = next_rg
19         if cur_rg >= M:
20             break
21     if segment[0] <= cur_rg and segment[1] > next_rg:
22         next_rg = segment[1]
23         cur_best = segment
24
25 if cur_rg < M:
26     cur_rg = next_rg
27     answer.append(cur_best)
28
29 if cur_rg < M:
30     print('No solution')
31 else:
32     print(len(answer))
33     print(*[str(a) + ' ' + str(b) for a, b in answer], sep='\n')
```

Отправить

Предыдущая

Следующая