Антон Макеев

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

9 окт 2021, 12:40:13
старт: 23 сен 2021, 12:00:00
начало: 23 сен 2021, 12:00:00

C. 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 ($1 \le M \le 5000$). В каждой последующей строке записана пара чисел L_i и R_i ($L_i, R_i \le 50000$), 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	1 0 1

```
0 0
```

Язык Swift 5.3

Набрать здесь | Отправить файл

```
import Foundation

func readIntArray() -> [Int] {
readLine()!
    .components(separatedBy: " ").compactMap { Int($0) }
}

let m = Int(readLine()!)!
var input = readIntArray()
var segments = [(start: Int, end: Int)]()

while input != [0, 0] {
    if input[1] > 0 && input[0] < m {
        segments.append((input[0], input[1]))
    }
    input = readIntArray()
}

segments.sort { $0.start < $1.start }

var nowRight = 0
var maxSegment = (start: 0, end: 0)
var coveringSegments = [(start: Int, end: Int)]()

for segment in segments {
    if segment.start > nowRight {
     coveringSegments.append(maxSegment)
        nowRight = maxSegment.end
        if nowRight < segment.start {
            break
        }
    }
    if maxSegment.end < segment.end {
        maxSegment = segment
    }
}

if nowRight < m {
```

Отправить

Предыдущая Следующая