

Антон Макеев

Яндекс. Тренировки по алгоритмам 2.0, занятие 5 (В)

9 окт 2021, 12:26:45

старт: 16 сен 2021, 12:00:00

начало: 16 сен 2021, 12:00:00

Е. Сумма трёх

Ограничение времени	15 секунд
Ограничение памяти	256Mb
Ввод	стандартный ввод или threesum.in
Вывод	стандартный вывод или threesum.out

Даны три массива целых чисел A, B, C и целое число S .
Найдите такие i, j, k , что $A_i + B_j + C_k = S$.

Формат ввода

На первой строке число S ($1 \leq S \leq 10^9$). Следующие три строки содержат описание массивов A, B, C в одинаковом формате: первое число задает длину n соответствующего массива ($1 \leq n \leq 15\,000$), затем заданы n целых чисел от 1 до 10^9 — сам массив.

Формат вывода

Если таких i, j, k не существует, выведите единственное число -1 . Иначе выведите на одной строке три числа — i, j, k . Элементы массивов нумеруются с нуля. Если ответов несколько, выведите лексикографически минимальный.

Пример 1

Ввод Вывод

3
2 1 2
2 3 1
2 3 1

0 1 1

Пример 2

Ввод Вывод

10
1 5
1 4
1 3

-1

Пример 3

Ввод

Вывод

5
4 1 2 3 4
3 5 2 1
4 5 3 2 2

0 1 2

Язык

Swift 5.3

Набрать здесь

Отправить файл

```
1 import Foundation
2
3 extension Array {
4     func removingFirst() -> Self {
5         var array = self
6         array.removeFirst()
7         return array
8     }
9 }
10
11 func readIntArray() -> [Int] {
12     readLine()?.trimmingCharacters(in: .whitespaces).components(separatedBy: " ").map { Int($0)! }
13 }
14
15 let tupleSort: ((Int,Int), (Int, Int)) -> Bool =
16     { $0.1 < $1.1 || $0.1 == $1.1 && $0.0 < $1.0 }
17
18 let numToFind = readIntArray()[0]
19 let firstSeq = readIntArray().removingFirst()
20 let secondSeq = readIntArray().removingFirst().enumerated().sorted(by: tupleSort)
21 let thirdSeq = readIntArray().removingFirst().enumerated().sorted(by: tupleSort)
22
23 func findIndicesForSum() -> (Int, Int, Int)? {
24
25     var ans: (i: Int, j: Int, k: Int)? = nil
26     var i = 0
27
28     while i < firstSeq.count && ans == nil {
29         let leftToFind = numToFind - firstSeq[i]
30         var j = 0
31         var k = (thirdSeq.firstIndex(where: { $0.element >= leftToFind }) ?? thirdSeq.count) - 1
32         while k >= 0 && j < secondSeq.count {
33             let currentSum = secondSeq[j].element + thirdSeq[k].element
34             if currentSum < leftToFind {
35                 j += 1
36             } else if currentSum > leftToFind {
37                 k -= 1
38             } else {
```

Отправить

Предыдущая