

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

⌚ 4 апр 2024, 02:39:02

старт: 28 окт 2023, 21:14:17

финиш: 26 ноя 2023, 03:14:16

длительность: 28д. 5ч.

...

Объявления жюри

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

[Положение участников](#) [Задачи](#) [Посылки](#)

6. Операционные системы lite

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Васин жесткий диск состоит из M секторов. Вася последовательно устанавливал на него различные операционные системы следующим методом: он создавал новый раздел диска из последовательных секторов, начиная с сектора номер a_i и до сектора b_i включительно, и устанавливал на него очередную систему. При этом, если очередной раздел хотя бы по одному сектору пересекается с каким-то ранее созданным разделом, то ранее созданный раздел «затирается», и операционная система, которая на него была установлена, больше не может быть загружена. Напишите программу, которая по информации о том, какие разделы на диске создавал Вася, определит, сколько в итоге работоспособных операционных систем установлено и работает в настоящий момент на Васином компьютере.

Формат ввода

Сначала вводятся натуральное число M — количество секторов на жестком диске ($1 \leq M \leq 10^9$) и целое число N — количество разделов, которое последовательно создавал Вася ($0 \leq N \leq 1000$). Далее идут N пар чисел a_i и b_i , задающих номера начального и конечного секторов раздела ($1 \leq a_i \leq b_i \leq M$).

Формат вывода

Выведите одно число — количество работающих операционных систем на Васином компьютере.

Пример 1

Ввод 	Вывод 
10 3 1 3 4 7 3 4	1

Пример 2

Ввод 	Вывод 
10 4 1 3 4 5 7 8 4 6	3

Язык

Python 3.9 (PyPy 7.3.11) ▾

Набрать здесь

Отправить файл

```
1 fin = open('input.txt')
2 M = int(fin.readline())
3 N = int(fin.readline())
4
5 os_partitions = [None]*N
6 os_valid = [True]*N
7
8 for i in range(N):
9     os_partitions[i] = [int(x) for x in fin.readline().split()]
10
11 for i in range(N-1, -1, -1):
12     for j in range(i-1, -1, -1):
13         if os_valid[j]: os_valid[j] = (os_partitions[i][1] < os_partitions[j][0] or os_partitions[i][0] < os_partitions[j][1])
14
15 valid_count = 0
16 for i in range(N):
17     valid_count += os_valid[i]
18 print(valid_count)
```

Отправить

i осталось 100 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
16 ноя 2023, 02:42:43	97484879	6	Python 3.9 (PyPy 7.3.11)	OK	-	220ms	28.10Mb	-	- отчёт
16 ноя 2023, 02:41:44	97484839	6	Python 3.9 (PyPy 7.3.11)	RE	-	243ms	28.08Mb	1	- отчёт
16 ноя 2023, 02:40:54	97484794	6	Python 3.9 (PyPy 7.3.11)	OK	-	228ms	28.09Mb	-	- отчёт
15 ноя 2023, 22:03:41	97457838	6	Python 3.9 (PyPy 7.3.11)	WA	-	210ms	28.09Mb	15	- отчёт
15 ноя 2023, 20:23:59	97444547	6	Python 3.9 (PyPy 7.3.11)	WA	-	195ms	28.10Mb	4	- отчёт

- ✓

1. Гистограмма
- ✓

2. Красивая строка
- ✓

3. Коллекционер Диего
- ✓

4. Контрольная работа
- ✓

5. Хорошая строка
- ✓

6. Операционные системы lite
- ✓

7. SNTP
- ✓

8. Минимальный прямоугольник
- ✓

9. Сумма в прямоугольнике
- ✓

10. Скучная лекция
- ✓

11. Стек с защитой от ошибок
- ✓

12. Правильная скобочная последовательность
- ✓

13. Постфиксная запись
- ✓

14. Сортировка вагонов lite
- ✓

15. Великое Лайнландское переселение
- ✓

16. Очередь с защитой от ошибок
- ✓

17. Игра в пьяницу
- ✓

18. Дек с защитой от ошибок
- ✓

19. Хипуй
- ✓

20. Пирамидальная сортировка
- ✗

21. Три единицы подряд
22. Кузнечик
23. Калькулятор
24. Покупка билетов
25. Гвоздики
26. Самый дешевый путь
27. Вывести маршрут максимальной стоимости
28. Ход конём
29. Кафе
30. НОП с восстановлением ответа
- ✓

31. Поиск в глубину
- ✓

32. Компоненты связности
- ✓

33. Списывание
- ✓

34. Топологическая сортировка
- ✓

35. Поиск цикла
- ✓

36. Длина кратчайшего пути
- ✓

37. Путь в графе
- ✓

38. Блохи
- ✓

39. Путь спелеолога
- ✓

40. Метро