

Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

4 апр 2024, 02:39:09
старт: 28 окт 2023, 21:14:17
финиш: 26 ноя 2023, 03:14:16
длительность: 28д. 5ч.
...

Объявления жюри

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

Положение участников Задачи Псылки

7. SNTP

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

Для того чтобы компьютеры поддерживали актуальное время, они могут обращаться к серверам точного времени SNTP (Simple Network Time Protocol). К сожалению, компьютер не может просто получить время у сервера, потому что информация по сети передаётся не мгновенно: пока сообщение с текущим временем дойдёт до компьютера, оно потеряет свою актуальность. Протокол взаимодействия клиента (компьютера, запрашивающего точное время) и сервера (компьютера, выдающего точное время) выглядит следующим образом:

1. Клиент отправляет запрос на сервер и запоминает время отправления A (по клиентскому времени).

2. Сервер получает запрос в момент времени B (по точному серверному времени) и отправляет клиенту сообщение, содержащее время B.

3. Клиент получает ответ на свой запрос в момент времени C (по клиентскому времени) и запоминает его. Теперь клиент, из предположения, что сетевые задержки при передаче сообщений от клиента серверу и от сервера клиенту одинаковы, может определить и установить себе точное время, используя известные значения A, B, C.

Вам предстоит реализовать алгоритм, с точностью до секунды определяющий точное время для установки на клиенте по известным A, B и C. При необходимости округлите результат до целого числа секунд по правилам арифметики (в меньшую сторону, если дробная часть числа меньше 1/2, иначе в большую сторону).

Возможно, что, пока клиент ожидал ответа, по клиентскому времени успели наступить новые сутки, однако известно, что между отправкой клиентом запроса и получением ответа от сервера прошло менее 24 часов.

Формат ввода

Программа получает на вход три временные метки A, B, C, по одной в каждой строке. Все временные метки представлены в формате «hh:mm:ss», где «hh» – это часы, «mm» – минуты, «ss» – секунды. Часы, минуты и секунды записываются ровно двумя цифрами каждое (возможно, с дополнительными нулями в начале числа).

Формат вывода

Программа должна вывести одну временную метку в формате, описанном во входных данных, – вычисленное точное время для установки на клиенте. В выводе не должно быть пробелов, пустых строк в начале вывода.

Пример

Ввод	Вывод
15:01:00	18:10:05
18:09:45	
15:01:40	

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь

Отправить файл

```
1 MAX_SECS_IN_DAY = 86400 # 24*60*60
2
3 def timeToTimeInSec(hours, minutes, seconds):
4     return hours*3600+minutes*60+seconds
5
6 def timeInSecToTime(time_in_secs):
7     hours = time_in_secs//3600
8     minutes = (time_in_secs-hours*3600)//60
9     seconds = time_in_secs-hours*3600-minutes*60
10
11     return f'{hours:0>2}:{minutes:0>2}:{seconds:0>2}'
12
13 def normalizeTimeInSecsToDay(time_in_secs):
14     if time_in_secs >= 0:
15         return time_in_secs % MAX_SECS_IN_DAY
16     else:
17         while time_in_secs < 0:
18             time_in_secs += MAX_SECS_IN_DAY
19         return time_in_secs
20
21 fin = open('input.txt')
22
23 h1, m1, s1 = [int(x) for x in fin.readline().split(':')]
24 h2, m2, s2 = [int(x) for x in fin.readline().split(':')]
25 h3, m3, s3 = [int(x) for x in fin.readline().split(':')]
26
27 time_s1 = timeToTimeInSec(h1, m1, s1)
28 time_s2 = timeToTimeInSec(h2, m2, s2)
29 time_s3 = timeToTimeInSec(h3, m3, s3)
30
31 send_time = ((time_s3-time_s1))/2
32 gap = (time_s2-time_s1-send_time)
33 exact_time = int(normalizeTimeInSecsToDay(time_s3+gap)+0.5)
34
35 print(timeInSecToTime(exact_time))
```

Отправить

осталось 100 попытки

Предыдущая

Следующая

Время послыски	ID	Задача	Компилятор	Вердикт	Тип послыски	Время	Память	Тест	Баллы	
16 ноя 2023, 07:08:04	97489596	7	Python 3.9 (PyPy 7.3.11)	WA	-	191ms	28.09Mb	9	-	отчёт
16 ноя 2023, 07:07:21	97489589	7	Python 3.9 (PyPy 7.3.11)	OK	-	189ms	28.10Mb	-	-	отчёт
16 ноя 2023, 06:49:55	97489377	7	Python 3.9 (PyPy 7.3.11)	WA	-	200ms	28.09Mb	9	-	отчёт
16 ноя 2023, 06:49:24	97489370	7	Python 3.9 (PyPy 7.3.11)	WA	-	191ms	28.09Mb	9	-	отчёт
16 ноя 2023, 06:43:45	97489290	7	Python 3.9 (PyPy 7.3.11)	WA	-	189ms	28.10Mb	9	-	отчёт
16 ноя 2023, 06:42:02	97489258	7	Python 3.9 (PyPy 7.3.11)	WA	-	217ms	28.09Mb	9	-	отчёт
16 ноя 2023, 06:41:11	97489251	7	Python 3.9 (PyPy 7.3.11)	WA	-	196ms	28.09Mb	9	-	отчёт
16 ноя 2023, 06:04:49	97488837	7	Python 3.9 (PyPy 7.3.11)	WA	-	194ms	28.09Mb	3	-	отчёт
16 ноя 2023, 06:03:31	97488816	7	Python 3.9 (PyPy 7.3.11)	WA	-	198ms	28.09Mb	3	-	отчёт
16 ноя 2023, 06:01:10	97488795	7	Python 3.9 (PyPy 7.3.11)	WA	-	188ms	28.09Mb	3	-	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 дешёвый 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