



Сдать решение задачи C

Полный балл:	1
Бонусные баллы:	
Ограничение времени:	1 с
Ограничение реального времени:	5 с
Ограничение памяти:	64М

Притормози, зеленоглазое такси!

При работе с API внешних сервисов стоит учитывать их пропускную способность. Одним из самых распространенных ограничений может быть лимит на количество запросов в секунду (rps).

Напишите декоратор "балансировщик нагрузки" @timeout(rps), который бы позволял вызывать функцию не более rps раз в 1 секунду.

Примеры

Входные данные

```
import time

@timeout(rps=5)
def func():
    pass

if __name__ == "__main__":
    t_start = time.time()
    for i in range(10):
        func()
    t_delta = time.time() - t_start
    print("{:.2f}".format(t_delta))
```

Результат работы

```
2.00
```

Входные данные

```
import random
import time

@timeout(rps=5)
def func():
    time.sleep(random.random() * 0.1)

if __name__ == "__main__":
    t_start = time.time()
    for i in range(10):
        func()
    t_delta = time.time() - t_start
    print("{:.2f}".format(t_delta))
```

Результат работы

```
2.00
```

Входные данные

```
import time

@timeout(rps=5)
def func():
    time.sleep(0.3)

if __name__ == "__main__":
    t_start = time.time()
    for i in range(10):
        func()
    t_delta = time.time() - t_start
    print("{:.2f}".format(t_delta))
```

Результат работы

```
3.00
```

Примечания

Не забудьте перекрыть имя декорируемой функции через декоратор из модуля functools.

Функция, обернутая в декоратор, должна возвращать то же значение, что и функция без декоратора. В задаче НЕ гарантируется, что оборачиваемая функция не принимает на вход аргументов.

Считайте, что каждый запуск должен выполняться не быстрее чем 2.1 / rps

Считайте, что каждый запуск должен выполняться не быстрее, чем за 1 / 1 рз.

Сдать решение

Язык: python3 - Python3 3.6.5

Файл  No file chosen

Отправить!

Предыдущие решения этой задачи

Номер решения	Время	Размер	Задача	Язык	Результат	Пройдено тестов	Баллы	Посмотреть текст	исходный	Посмотреть протокол
647	135:54:57	431	C	python3	OK	6	1	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>
641	127:23:39	1142	C	python3	Неполное решение	5	0	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>
639	127:20:13	728	C	python3	Неполное решение	5	0	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>
632	127:09:46	956	C	python3	Неполное решение	5	0	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>
631	126:59:23	647	C	python3	Неполное решение	4	0	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>
627	125:59:53	427	C	python3	Неполное решение	2	0	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>
623	125:55:20	426	C	python3	Неполное решение	2	0	<a href="#">Посмотр</a>		<a href="#">Посмотр</a>

- A

B

C

D

E

F

G

H

I