

Контест для Weekend offer backend

🕒 30 май 2021, 12:06:06

🏁 старт: 30 май 2021, 11:05:30

🏁 финиш: 30 май 2021, 13:05:30

🕒 до финиша: 00:59:17

...

Объявления жюри

Завершить

Задачи [Посылки](#)

Е. Ограничение запросов [2 балла]

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	3 секунды	256Mb	стандартный ввод	стандартный вывод
Golang 1.14.4 + network	2 секунды	256Mb		
Python 3.7 + network + requests	5 секунд	256Mb		
GNU c++ 11 + network + libcurl + json	2 секунды	256Mb		

Слава только-только устроился в Яндекс.Маркет и получил первое задание: написать компонент, отвечающий за ограничение запросов пользователей к сервису.

Характеристики компонента — лимит запросов на пользователя, лимит запросов на сервис и продолжительность интервала (*duration*), за который учитываются запросы. Запрос к сервису, поступивший в момент времени *time*, отклоняется, если он удовлетворяет хотя бы одному из двух условий:

- Если в промежутке времени $[time - duration, time]$ от данного пользователя было принято к выполнению не меньше пользовательского лимита запросов — должен возвращаться код ошибки «Too Many Requests».
- Если в промежутке времени $[time - duration, time]$ суммарно от всех пользователей было принято к выполнению не меньше сервисного лимита запросов — должен возвращаться код ошибки «Service Unavailable».

Условия проверяются последовательно друг за другом, пока не будет найдено первое условие, которому запрос удовлетворяет. Если запрос не удовлетворяет ни одному условию, то он выполняется. Пока что Слава еще не до конца погрузился в разработку и внутренние технологии, поэтому решил написать прототип, а вы можете помочь Славе.

Формат ввода

Программа получает на вход строку с тремя числами, разделенными пробелами: *userLimit* ($1 \leq userLimit \leq 5 \cdot 10^4$) — лимит запросов на пользователя, *serviceLimit* ($1 \leq serviceLimit \leq 5 \cdot 10^4$) — лимит запросов на сервис, *duration* ($1 \leq duration \leq 10^9$) — промежуток времени в миллисекундах, за который учитываются запросы для расчета лимитов.

В следующих строках поступают описания запросов. Каждый запрос представлен двумя разделенными пробелом числами: *time* ($1 \leq time \leq 10^9$) — время поступления запроса, *userId* ($1 \leq userId \leq 10^9$) — идентификатор пользователя, выполнившего запрос.

Входные данные завершаются строкой с числом —1. Гарантируется, что все времена в описании запросов не убывают, а количество запросов не превышает $5 \cdot 10^4$.

Тестирующая система даст прочитать очередной запрос только после записи в стандартный вывод ответа на предыдущий запрос (не забудьте после вывода запроса сделать `flush`).



Формат вывода

Для каждого запроса необходимо вывести один из трех кодов ответа HTTP:

- 200 — если запрос будет выполнен.
- 429 — если запрос будет отклонен из-за превышения лимита запросов пользователем.
- 503 — если запрос будет отклонен из-за превышения сервисного лимита запросов.

Не забывайте о том, что ваша программа должна сбрасывать буфер вывода после вывода строки с тайм-аутом. Для сброса буфера вывода можно использовать `fflush(stdout)` в C++, `System.out.flush()` в Java, `stdout.flush()` в Python.

Пример

Ввод 	Вывод 
2 5 5	200
1 100	200
1 100	429
2 100	200
2 200	200
2 300	200
2 400	503
2 500	503
3 500	503
5 200	429
6 100	200
7 200	
−1	

Язык

Java 8 + network + json-... 

Набрать здесь

Отправить файл




1

Отправить

 осталось 100 попыток

Предыдущая

 Посылка нет

-  [А. Камни и украшения. \[0 баллов\]](#)
- [В. Разбиение на интервалы дат \[1 балл\] \(1\)](#)
- [С. Опять JSON'ы перекладывать... \[1 балл\] \(1\)](#)
-  [D. Сортировка чисел \[1 балл\] \(1\)](#)
-  [Е. Ограничение запросов \[2 балла\] \(2\)](#)