

# Тренировки по алгоритмам 5.0 от Яндекса — Занятие 3 (Множества и словари)

28 мар 2024, 20:14:31  
старт: 15 мар 2024, 22:30:00  
финиш: 27 мар 2024, 20:00:00  
длительность: 11д. 21ч.  
начало: 15 мар 2024, 22:30:00  
конец: 27 мар 2024, 20:00:00

## J. P2P обновление

Ограничение времени	15 секунд
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В системе умного дома под управлением голосового помощника Лариса  $n$  устройств, соединяющихся между собой по сети LoRaWAN. Устройство номер 1 подключено к интернету и на него было скачано обновление, которое необходимо передать на все устройства. Сеть LoRaWAN очень медленная, поэтому для распространения протокола был придуман peer-to-peer (P2P) протокол. Файл обновления разбивается на  $k$  одинаковых по размеру частей, занумерованных от 1 до  $k$ .

Передача части обновления происходит во время таймслотов. Каждый таймслот занимает одну минуту. За один таймслот каждое устройство может получить и передать ровно одну часть обновления. То есть устройство во время таймслота может получать новую часть обновления и передавать уже имеющуюся у него к началу таймслота часть обновления, или совершать только одно из этих действий, или вообще не осуществлять прием или передачу. После приема части обновления устройство может передавать эту часть обновления другим устройствам в следующих таймслотах.

Перед каждым таймслотом для каждой части обновления определяется, на скольких устройствах сети скачана эта часть. Каждое устройство выбирает отсутствующую на нем часть обновления, которая встречается в сети реже всего. Если таких частей несколько, то выбирается отсутствующая на устройстве часть обновления с наименьшим номером.

После этого устройство делает запрос выбранной части обновления у одного из устройств, на котором такая часть обновления уже скачана. Если таких устройств несколько — выбирается устройство, на котором скачано наименьшее количество частей обновления. Если и таких устройств оказалось несколько — выбирается устройство с минимальным номером.

После того, как все запросы отправлены, каждое устройство выбирает, чей запрос удовлетворить. Устройство  $A$  удовлетворяет тот запрос, который поступил от наиболее ценного для  $A$  устройства. Ценность устройства  $B$  для устройства  $A$  определяется как количество частей обновления, ранее полученных устройством  $A$  от устройства  $B$ . Если на устройство  $A$  пришло несколько запросов от одинаково ценных устройств, то удовлетворяется запрос того устройства, на котором меньше всего скачанных частей обновления. Если и таких запросов несколько, то среди них выбирается устройство с наименьшим номером.

Далее начинается новый таймслот. Устройства, чьи запросы удовлетворены, скачивают запрошенную часть обновления, а остальные не скачивают ничего.

Для каждого устройства определите, сколько таймслотов понадобится для скачивания всех частей обновления.

### Формат ввода

Вводится два числа  $n$  и  $k$  ( $2 \leq n \leq 100, 1 \leq k \leq 200$ ).

### Формат вывода

Выведите  $n-1$  число — количество таймслотов, необходимых для скачивания обновления на устройства с номерами от 2 до  $n$ .

### Пример

Ввод  Вывод

3 2 3 3

# Примечания

Для удобства будем пользоваться обозначениями устройств буквами А, В, С (соответствует устройствам с номерами 1, 2 и 3). На устройстве А есть обе части обновления, а на устройствах В и С — ни одной.

Перед первым таймслотом для каждой части определяется количество устройств, на которых скачана каждая часть обновления: и 1 и 2 часть обновления присутствуют только на одном устройстве.

Устройства В и С выбирают самую редкую отсутствующую у них часть обновления с минимальным номером: самая редкая часть с минимальным номером — это часть 1. Она отсутствует и на устройстве В, и на устройстве С. Они запрашивают ее у устройства А. Ценность устройств В и С для устройства А равна нулю. Количество имеющихся у устройств В и С частей обновления одинакова и равно нулю. Поэтому устройство А выбирает устройство с минимальным номером (В). Во время первого таймслота выполняется передача части 1 с устройства А на устройство В. Ценность устройства А для устройства В становится равной 1.

Перед вторым таймслотом для каждой части определяется количество устройств, на которых скачана каждая часть обновления: самой редкой оказывается часть 2 (присутствует только на устройстве А), следующая по редкости часть 1 (присутствует на устройствах А и В).

Устройства В и С выбирают среди отсутствующих у них частей обновления самую редкую: для обоих устройств выбирается часть 2. Каждое из них делает запрос части 2 у единственного обладателя этой части — устройства А. Ценность устройств В и С для устройства А одинакова и равна нулю. Количество имеющихся у устройства С частей (0) меньше, чем у устройства В (1), поэтому выбирается устройство С. Во время второго таймслота выполняется передача части 2 с устройства А на устройство С. Ценность устройства А для устройства С становится равной 1.

Перед третьим таймслотом для каждой части определяется количество устройств, на которых скачана каждая часть обновления: обе части 1 и 2 присутствуют на двух устройствах (часть 1 на устройствах А и В, часть 2 — на устройствах А и С)

Устройство В может сделать запрос недостающей части 2 у обладающей ей устройств А и С, но выбирает устройство С, т.к. на устройстве С скачано меньше частей (1), чем у устройства А (2).

Устройство С может сделать запрос недостающей части 1 у обладающей ей устройств А и В, но выбирает устройство В, т.к. на устройстве В скачано меньше частей (1), чем у устройства А (2).

Во время третьего таймслота оба запроса оказываются единственными запросами у устройств В и С и удовлетворяются. Часть 2 передается с устройства С на устройство В. Часть 1 передается с устройства В на устройство С. Ценность устройства В для устройства С становится равной 1. Ценность устройства С для устройства В становится равной 1.

Все части обновления оказываются на всех устройствах и на этом обновление заканчивается.

Язык 

Kotlin 1.9.21 (JRE 21)

Набрать здесь

Отправить файл

1

Отправить

Предыдущая