Задача сетевой связанности на графах

Представим, что в городе N есть мобильная сеть. Строительство нового района всегда начинается с возведения вышки мобильной связи. Все пользователи района подключаются к этой вышке, чтобы иметь возможность пользоваться услугами сети (отправлять получать сообщения). Таким образом, независимо от того, насколько велик город, все пользователи могут отправлять сообщения друг другу пока все вышки подключены между собой (между ними есть сетевая связанность).

Одна из главных целей мэра города N - контролировать мобильную связанность граждан города. Показатель мобильной связанности города это сумма мобильной связанности всех граждан города. В свою очередь, показатель мобильной связанности каждого гражданина равен количеству граждан (всегда включая себя), которым можно отправлять сообщения.

Поскольку город N растет, граждане решили, что мэру нужен помощник, чтобы тот мог сосредоточиться на защите вышек. Основная задача помощника выяснить, какие вышки следует защищать первыми. **Основная цель выбрать самую важную вышку в сети. Если несколько вышек имеют максимальную важность, надо обозначить их все. Важность вышки определяется ее влиянием на показатель мобильной связанности города** чем меньше показатель после уничтожения вышки, тем она важнее.

Запуск программы в ОС Linux

Для запуска программы в ОС Linux достаточно лишь написать в терминале python3 <полный путь до файла с программой> и нажать Enter.

Файл, содержащий входные данные, должен иметь формат .txt, называться input.txt и находиться в одной директории с запускаемой программой.

Для корректной работы программы содержимое файла должно иметь чёткую струтуру:

$$\begin{gathered}\text{Name}_1\ \text{Name}_2\\ \dots\\ \text{Name}_{n-1}\ \text{Name}_n\\ <\text{пустая строка}>\\ \text{Name}_1\ \text{Users}_1\\ \dots\\ \text{Name}_k\ \text{Users}_k\end{gathered}$$

То есть входные данные это два множества: 1 - список соединений между вышками (каждое соединение пишется на отдельной строке, имена соединённых вышек разделяются пробелом; в именах вышек не должны содержаться пробельные символы), 2 - список вышек с количеством пользователей (каждая вышка описывается один раз на отдельной строке; сначала записывается имя, затем через пробел одно целое число). Сами множества разделяются пустой строкой.

В противном случае программа преждевременно завершится и постарается указать на ошибку.

Запуск программы в Play with Docker

Зайдя на сайт https://labs.play-with-docker.com/ нажмите Start, затем слева +ADD NEW INSTANCE, появится окно терминала. В него нужно перетащить 4 файла, приложенные к письму, - start.sh, continue.sh, stop.sh и input.txt. Убедиться, что они добавлены, можно, нажав на "EDITOR"над окном терминала.

Первые 3 файла нужно сделать исполняемыми. Я предлагаю 2 способа. Либо выполнить 3 команды:

```
CHMOD +X START.SH
CHMOD +X CONTINUE.SH
CHMOD +X STOP.SH
```

и затем запускать эти файлы так:

```
./START.SH
./CONTINUE.SH
./STOP.SH
```

Либо просто запускать файлы так:

```
SH START.SH
SH CONTINUE.SH
SH STOP.SH
```

Эти 2 способа являются эквивалентными. После выполнения start.sh можно любое количество раз выполнить continue.sh, после выполнения stop.sh, для продолжения тестирования необходимо заново выполнить start.sh.

Прежде чем выполнять файл continue.sh нужно изменить файл input.txt. Делается это так: нажать на "EDITOR там дважды кликнуть на input.txt, внести изменения в открывшемся окошке, нажать Save. После этого можно выполнять continue.sh. Обратите внимание на то, что входной файл input.txt должен иметь структуру, описанную в предыдущем разделе. Результат работы сервера всегда выводится последней строкой после выполнения start.sh и continue.sh.

Проверить соблюдение условий задания можно, просмотрев содержимое файлов start.sh, continue.sh, stop.sh. Для тестирования на локальном компьютере необходимо в файлах start.sh и continue.sh изменить путь к файлу input.txt в строке запуска клиентского контейнера.