Сегментирование пользователей по IP адресам.

Задача

Сегментом будем называть множество идентификаторов пользователей (userId).

На основе файла транзакций и файла диапазонов IP адресов необходимо провести сегментирование пользователей, т.е. определить каким сегментам они будут принадлежать

Вход

На вход поступают два файла:

1. **"Файл транзакций"**. Состоит из строк вида:
   <userId>символ_табуляции<IP адрес>.

   Пример:
   6546263252026003359 192.168.1.1

   Размер файла транзакций не ограничен. UserId могут повторяться с разными IP.

2. **"Файл диапазонов"**. Состоит из строк вида:
   <минимальный IP адрес>-<максимальный IP адрес>символ_табуляции<название сегмента>.
   Пример:
   192.168.1.0-192.168.1.255 SegmentName

   Можно считать, что "минимальный IP адрес" <= "максимальный IP адрес". Диапазоны могут пересекаться. Одному сегменту может соответствовать несколько диапазонов адресов.
   Обычный уровень - размер файла диапазонов 100 строк.
   Продвинутый уровень - размер файла диапазонов 1 000 000 строк.

где
- userId - это знаковое 64-битное число (Long), например: -8134683252591960481, 6546263252026003359
- IP адрес - строковое представление IPv4 адреса (четыре числа от 0 до 255, разделённых точками). Например: 1.2.3.4, 0.0.0.0, 255.255.0.0, 192.168.1.1

- название сегмента - строка, состоящая из латинских символов и/или цифр ([a-zA-Z0-9]+)

Выход

В качестве выхода ожидается текстовый файл output.tsv, состоящий из строк вида:
<userId>символ_табуляции<название сегмента>

Один userId может принадлежать нескольким сегментам. В этом случае ожидается на выход несколько строк.

Пример

transactions.tsv
1 192.168.1.1
11 192.168.1.2
2 192.168.2.2
3 192.168.3.3

ranges.tsv
192.168.1.0-192.168.1.255 Network1
192.168.2.0-192.168.2.255 Network2
192.168.3.0-192.168.3.255 Network3

output.tsv:
1 Network1
11 Network1
2 Network2
3 Network3

Дополнительные требования

- (!) Приветствуются наиболее оптимальные методы решения задачи (но это не является обязательным условием выполнения задания).
- В качестве языка программирования очень желательно использовать Scala; в качестве инструмента сборки SBT.
- Вне зависимости от используемого для решения языка, среда разработки (Scala, SBT) должна быть настроена до начала занятий. Ожидается версия Scala = 2.11.7. Java 8, sbt > 0.13
- При использовании Scala результатом работы должен быть sbt проект: будет вызываться комманда "sbt run" для тестового запуска приложения.

Полезные ссылки

- https://twitter.github.io/scala_school/ (Введение в scala)
- http://www.scala-sbt.org/0.13/docs/Getting-Started.html (Поможет настроить SBT)
- http://docs.scala-lang.org/
- http://www.scala-lang.org/files/archive/api/2.11.7/index.html#scala.io.Source (Быстрый способ работы с IO)
- https://github.com/paulp/sbt-extras (Для продвинутых SBT пользователей :-))