

Результат работы алгоритма:

1 вариант - структура с ключами ip и значениями - массив временных отметок. Плюсы - требует меньше памяти, минусы - специфическая структура данных.

2 вариант - таблица с логами, соответствующими отметкам. Плюсы - унификация, на вход таблица, на выход таблица, минусы - больший размер в сравнении с 1-м вариантом.

Начало алгоритма

Вход - таблица логов, параметр точности accuracyTime создания ip блока - неотрицательное целое число минут.

Берем строку из входной таблицы, если координата условно равна текущей, то ничего не меняем, берем следующую строку.

Если нет, то запускаем модуль изменения координаты.

Есть предыдущая координата, текущая отличается. Если по времени отличается от предыдущей меньше, чем на accuracyTime, то берем следующий лог и т.д. Изменения происходят в следующих двух случаях - если координата вернулась к исходному значению, тогда запускается общая схема.

Если координата не равна исходному значению и отличается от зафиксированной на величину, большую, чем accuracyTime, то фиксированная точка считается концом ip-блока. Следующий за ним лог считается началом нового и алгоритм повторяется.

Data: logs base

Result: Scheme of common groups on logs' set

```
1 SELECT * FROM data WHERE data.coord IS NOT NULL ORDER BY
   ip, time;
2 while not end of data do
3   | cin >> line;
4   | compIP ← currentIP ← line.ip;
5   | tempCoord ← line.coord;
6   | if compIP = currentIP then
7   |   | Proc1;
8   | else
9   |   | currentIP = line.ip;
10  |   | Proc2;
11  | end
12 end
```

Algorithm 1: Group logs algorithm



