Amazon | OA 2020 | Transaction Logs

LEETCODE : <https://leetcode.com/discuss/interview-question/862600/>

Amazon parses logs of user transactions/activity to flag fraudulent activity. The log file is represented as an Array of arrays. The arrays consist of the following data:

[<userid1> <userid2> <# of transactions>]

For example:

[345366 89921 45]  
Note the data is space delimited

So, the log data will look like:

[

[345366 89921 45],

[029323 38239 23]

...

]

Write a function to parse the log data to find distinct user count that meets or crosses a certain threshold. The function will take in 2 inputs:

**Input 1:** Log data in form an array of arrays  
**Input 2:** threshold as an integer

Output should be an array of userids that are sorted.

If same userid appears in the transaction as userid1 and userid2, it should count as one occurence, not two.

Example:  
**Input 1:**

[

[345366 89921 45],

[029323 38239 23],

[38239 345366 15],

[029323 38239 77],

[345366 38239 23],

[029323 345366 13],

[38239 38239 23]

...

]

**Input 2**: 3

**Ouput:** [345366 , 38239, 029323]

Explanation:  
Given the following counts of userids, there are only 3 userids that meet or exceed the threshold of 3.  
345366 -4 , 38239 -5, 029323-3, 89921-1

Sol:

public class FraudLogs {

public List<String> getFraudIds(String[] input, int threshold) {

List<String> res = new ArrayList<>();

Map<String, Integer> map = new HashMap<>();

for (String log : input) {

String[] logVals = log.split(" ");

Set<String> set = new HashSet<>(Arrays.asList(logVals[0], logVals[1]));

for (String userId : set) {

map.put(userId, map.getOrDefault(userId, 0) + 1);

}

}

for (String userId : map.keySet()) {

if (map.get(userId) >= threshold)

res.add(userId);

}

Collections.sort(res);

return res;

}

public static void main(String[] args) {

String[] input = new String[] { "345366 89921 45", "029323 38239 23", "38239 345366 15", "029323 38239 77",

"345366 38239 23", "029323 345366 13", "38239 38239 23" };

System.out.println(new FraudLogs().getFraudIds(input, 3));

}

}