IDENTIFY FRAUDULENT TRANSACTIONS

Data Structure Project Development

3

Step 1

The first task is to design a “struct” to represent a credit card record. So, we could read tuples in a credit card record from the input data. Enter

Step 2

Next, continue to read all credit card records. We need to design a proper data structure to store the records read. An array will do the job nicely. When this stage is done, our program should output: the total number of credit card records read, the average daily limit per card (up to two digits after the decimal point, by using “%.2f” in printf()), and the credit card with the largest transaction limit (if there is a tie, print the card with the smallest ID). Enter

Step 3

Our third task is to design a struct to represent a transaction, read in the transactions, store them in a linked data structure, and output their IDs. We need to modify the linked list implementation in “listops.c” (link to source code available in lecture slides) to store the credit cards. Enter

Step 4

The last stage is to check whether a transaction may be fraudulent. We will go through the transactions. For each transaction, we need to check if it exceeds the transaction limit or the daily limit of the corresponding credit card. We will use the binary search algorithm (“binary_search()” function, link to source code available in lecture slides, or “bsearch()” provided by “stdlib.h”) to look up the credit card ID of each transaction from the credit card records read in Stage 2. Moreover, in order to reduce the time consumption of the program, we will only go through the transaction list once, that is, we need to design an algorithm with an O(nlogm) average time complexity for this stage, given n transactions and m credit card records. Enter

Introduction

There are around 16 million credit cards on issue in Australia,1 and the number is over 1 billion worldwide. This is a goldmine for cybercriminals who make unauthorised payment to obtain goods or services (i.e., credit card fraud). Worldwide losses from card fraud is expected to reach US$31 billion in 2020. Banks and card companies are strongly motivated to develop anti-fraud technologies. They prevented two-thirds of the attempted card fraud in the UK in 2018, but this is a never-ending battle. There are various anti-fraud algorithms. The core of those algorithms are rules and statistics (machine learning algorithms) to classify whether a transaction is abnormal and likely to be fraudulent. For example, a transaction well beyond the credit limit of a card is likely to be fraudulent, and so are two transactions of the same card issued at almost the same time but from two different cities. In this project, I wrote a program to process credit card and transaction records and identify fraudulent transactions. Enter

No Bug

No Risk

No Invision

OUTCOME

CODE: github.com/chenjiang0819/IdentifyFraudulentTransactions

```
Compiling with gcc -Wall -std=c99 -lm ...
Compilation succeeded.

==========================================
Test for input file: invis0.in
dewww0p11 100 100
eewww0p22 105 105
fewww0p66 150 100
%%%%%%%%%%
1yuy3noa2uxu fewww0p66 2020:05:07:04:16:20 72
9mopqy3snk3h fewww0p66 2020:05:07:08:06:49 86
gl3utnnwxf49 fewww0p66 2020:05:07:09:39:00 67
6hjqaydtmrq5 fewww0p66 2020:05:07:10:09:50 213
gl3utnnwxf40 fewww0p66 2020:05:07:11:39:00 67
mlgtqk8oo74e fewww0p66 2020:05:15:13:45:29 95

Expected results:                                   Your results:
=======================Stage 1========================      =======================Stage 1========================
Card ID: dewww0p11                                  Card ID: dewww0p11
Daily limit: 100                                    Daily limit: 100
Transaction limit: 100                              Transaction limit: 100

=======================Stage 2========================      =======================Stage 2========================
Number of credit cards: 3                           Number of credit cards: 3
Average daily limit: 118.33                         Average daily limit: 118.33
Card with the largest transaction limit: eewww0p22  Card with the largest transaction limit: eewww0p22

=======================Stage 3========================      =======================Stage 3========================
1yuy3noa2uxu                                        1yuy3noa2uxu
9mopqy3snk3h                                        9mopqy3snk3h
gl3utnnwxf49                                        gl3utnnwxf49
6hjqaydtmrq5                                        6hjqaydtmrq5
gl3utnnwxf40                                        gl3utnnwxf40
mlgtqk8oo74e                                        mlgtqk8oo74e

=======================Stage 4========================      =======================Stage 4========================
1yuy3noa2uxu          IN_BOTH_LIMITS                1yuy3noa2uxu          IN_BOTH_LIMITS
9mopqy3snk3h          OVER_DAILY_LIMIT              9mopqy3snk3h          OVER_DAILY_LIMIT
gl3utnnwxf49          OVER_DAILY_LIMIT              gl3utnnwxf49          OVER_DAILY_LIMIT
6hjqaydtmrq5          OVER_BOTH_LIMITS              6hjqaydtmrq5          OVER_BOTH_LIMITS
gl3utnnwxf40          OVER_DAILY_LIMIT              gl3utnnwxf40          OVER_DAILY_LIMIT
mlgtqk8oo74e          IN_BOTH_LIMITS                mlgtqk8oo74e          IN_BOTH_LIMITS

Your results seem to be CORRECT.  :)
```

```

==========================================
Test for input file: invis1.in
14f8iegn 300 200
1gs709c4 50 30
7feu9bll 1000 900
eg5lohwx 200 100
vc1ndc3o 205 111
%%%%%%%%%%
l25zo2b6jo2e 14f8iegn 2020:05:07:04:16:20 72
9n2rccovda70 1gs709c4 2020:05:07:08:06:49 16
i1oxihshjt41 7feu9bll 2020:05:07:09:39:00 17
un2ie4ag0af3 eg5lohwx 2020:05:07:10:09:50 21
l9e2gpdymu0n vc1ndc3o 2020:05:07:11:39:00 67
skjmoojd9zdj 14f8iegn 2020:05:07:13:45:29 201
87hv6tw75myd 1gs709c4 2020:05:07:15:44:01 31
a5myj9u9esk4 7feu9bll 2020:05:07:17:27:02 901
z2m54pdrcdob eg5lohwx 2020:05:07:18:28:03 101
sg82jkxyidon vc1ndc3o 2020:05:07:19:18:04 112
uevirknux6rj 14f8iegn 2020:05:07:22:40:05 847
9cp07pqdvszy 1gs709c4 2020:05:07:23:41:06 152
m9qv1icu3kwf 7feu9bll 2020:05:07:23:48:07 1870
u078aviozgka eg5lohwx 2020:05:07:23:54:08 236
uhuavz77l69n vc1ndc3o 2020:05:07:23:57:09 195

Expected results:                                   Your results:
=======================Stage 1========================      =======================Stage 1========================
Card ID: 14f8iegn                                   Card ID: 14f8iegn
Daily limit: 300                                    Daily limit: 300
Transaction limit: 200                              Transaction limit: 200

=======================Stage 2========================      =======================Stage 2========================
Number of credit cards: 5                           Number of credit cards: 5
Average daily limit: 351.00                         Average daily limit: 351.00
Card with the largest transaction limit: 7feu9bll   Card with the largest transaction limit: 7feu9bll

=======================Stage 3========================      =======================Stage 3========================
l25zo2b6jo2e                                        l25zo2b6jo2e
9n2rccovda70                                        9n2rccovda70
i1oxihshjt41                                        i1oxihshjt41
un2ie4ag0af3                                        un2ie4ag0af3
l9e2gpdymu0n                                        l9e2gpdymu0n
skjmoojd9zdj                                        skjmoojd9zdj
87hv6tw75myd                                        87hv6tw75myd
a5myj9u9esk4                                        a5myj9u9esk4
z2m54pdrcdob                                        z2m54pdrcdob
sg82jkxyidon                                        sg82jkxyidon
uevirknux6rj                                        uevirknux6rj
9cp07pqdvszy                                        9cp07pqdvszy
m9qv1icu3kwf                                        m9qv1icu3kwf
u078aviozgka                                        u078aviozgka
uhuavz77l69n                                        uhuavz77l69n

=======================Stage 4========================      =======================Stage 4========================
l25zo2b6jo2e          IN_BOTH_LIMITS                l25zo2b6jo2e          IN_BOTH_LIMITS
9n2rccovda70          IN_BOTH_LIMITS                9n2rccovda70          IN_BOTH_LIMITS
i1oxihshjt41          IN_BOTH_LIMITS                i1oxihshjt41          IN_BOTH_LIMITS
un2ie4ag0af3          IN_BOTH_LIMITS                un2ie4ag0af3          IN_BOTH_LIMITS
l9e2gpdymu0n          IN_BOTH_LIMITS                l9e2gpdymu0n          IN_BOTH_LIMITS
skjmoojd9zdj          OVER_TRANS_LIMIT              skjmoojd9zdj          OVER_TRANS_LIMIT
87hv6tw75myd          OVER_TRANS_LIMIT              87hv6tw75myd          OVER_TRANS_LIMIT
a5myj9u9esk4          OVER_TRANS_LIMIT              a5myj9u9esk4          OVER_TRANS_LIMIT
z2m54pdrcdob          OVER_TRANS_LIMIT              z2m54pdrcdob          OVER_TRANS_LIMIT
sg82jkxyidon          OVER_TRANS_LIMIT              sg82jkxyidon          OVER_TRANS_LIMIT
uevirknux6rj          OVER_BOTH_LIMITS              uevirknux6rj          OVER_BOTH_LIMITS
9cp07pqdvszy          OVER_BOTH_LIMITS              9cp07pqdvszy          OVER_BOTH_LIMITS
m9qv1icu3kwf          OVER_BOTH_LIMITS              m9qv1icu3kwf          OVER_BOTH_LIMITS
u078aviozgka          OVER_BOTH_LIMITS              u078aviozgka          OVER_BOTH_LIMITS
uhuavz77l69n          OVER_BOTH_LIMITS              uhuavz77l69n          OVER_BOTH_LIMITS

Your results seem to be CORRECT.  :)
```