Project Euler Problem 79 - By set theory and paper

*“*[*By analysing a user's login attempts, can you determine the secret numeric passcode?*](http://projecteuler.net/index.php?section=problems&id=79)*”*

*A common security method used for online banking is to ask the user for three random characters from a passcode. For example, if the passcode was 531278, they may ask for the 2nd, 3rd, and 5th characters; the expected reply would be: 317.*

*The text file,* [*keylog.txt*](http://projecteuler.net/project/keylog.txt)*, contains fifty successful login attempts.*

*Given that the three characters are always asked for in order, analyse the file so as to determine the shortest possible secret passcode of unknown length.*

So, for me, with absolutely no decryption/code-cracking experience this problem looks quite daunting at first sight. I started by saving the attached text file and importing the lines into my Python shell. Easy enough.

A general rule for problem solving is to consider the simplest case and work out the logic needed from there. This in mind, I chose to assume that all characters appeared once and only once.

Therefore all make a set of all possible digits (A) using Python’s wonderful set() object. This set contains only 8 digits (from the possible 10 digits 0 → 9). So under the distinct digit assumption and with the problem saying to find the shortest possible, the code is 8 digits long.

\*\*\*\*\*\*\*\*

We are told the characters are always listed in order so the characters that only ever occur in first in the data are candidates for the first character. My method was to make a new set of all characters that appeared in either second or third place (B). Clearly this is a subset:

And the compliment contains all first character candidates

This set actually contains only ‘7’ so:

7\*\*\*\*\*\*\*

This logic can also be applied to the second character. The set of all characters in third place (C) will never contain the second character.

But 7 is the first character so under the assumption we have:

73\*\*\*\*\*\*

Applying this reasoning to the last character: the set of all characters appearing in the first and second place in the data (D) can never contain the last digit:

So:

73\*\*\*\*\*0

And the set of all characters appearing in the first place in the data (E) can never contain the penultimate digit:

So we have

73\*\*\*\*90

At this point I couldn’t think up any more logic using sets distinct element sets so I looked at the data file with my eyes and continued on paper.

I noticed the following:

* 1 comes before 9 (no help)
* 1 comes before 6
* 6 comes before 8
* 2 comes before 8
* 6 comes before 2

Therefore:

73**1628**90

I figured hey, why not try it. It was correct.