

# シャフラーズ問題

© 2013 結城浩

<http://www.hyuki.com/codeiq/>

2013 年 5 月



## 1 概要

あなたはシャフラーズ星のプログラマとして雇われてきました。作業の内容は「壊れたユーザー一覧ファイルの復元」です。世界中のあちこちに残っていた断片的な id と name の集まりから、もとのファイルを可能な限り復元するのですが、残念ながら断片的な情報では、たくさんの name の集まりがシャッフルされています。シャッフルされた情報の中から、id と name の正しい対応をできるだけたくさん見つけ出すのがあなたのミッションなのです！

## 2 会話

あなた「仕事は壊れたユーザー一覧ファイルの復元作業と聞きましたが」

依頼者「順を追って説明しましょう。もともと我が社ではユーザの id と name の対応を以下のようにたくさん保持したファイルを扱っていました」

```
159 = joe
265 = her
314 = que
323 = mari
358 = asuka
846 = rei
979 = shinji
```

ユーザー一覧ファイルの例

あなた「なるほど。この “id = name” という形式で 1 個の対応を表しているのですね」

依頼者「そうです。id は固定桁の数字列で、name は英小文字列です。ファイルの各行に “id = name” が書かれていました。しかし、先日の突発性超パーミュテーション磁気嵐の影響ですべてがぐしゃぐしゃになってしまったのです」

あなた「突発性パーミュ……？」

依頼者「社内のファイルはすべて壊れてしまったのですが、さいわい、世界中に分散していた断片的な情報を集めることができたので、復元を試みることにしました。そうして、以下のような復元途中のファイルができたのです」

```
159 265 314 = her joe que
159 323 358 979 = asuka joe mari shinji
265 314 846 = her que rei
265 979 = her shinji
```

復元途中のファイル例

あなた「これはどう読めばよいのですか」

依頼者「ファイルの各行はこうなっています。1 個以上の id が並び、1 個のイコール (=) が続き、さらに 1 個以上の name が並びます。それらの要素はすべて 1 個のスペースで区切られています」

あなた「そこまでできているのなら完全な復元も簡単ですね。プログラムを書いて、両辺の id と name を正しく 1 個ずつ対応づけて、“id = name” の形にすればいいのでしょうか？ たとえば、159 265 314 = her joe que から、159 = her と 265 = joe と 314 = que を作ればいいですね」

依頼者「いえ、そう簡単ではありません。磁気嵐の影響で、世界各地から集めた断片的な情報では、id 列も name 列もシャッフルされてしまっていたからです」

あなた「シャッフルというのは、つまり、順序が入れ替わっているということでしょうか」

依頼者「そうです。たとえば 159 265 314 = her joe que に対して、id と name の正しい対応は以下の 1 から 6 のいずれかになります」

	159	265	314
1	her	joe	que
2	her	que	joe
3	joe	her	que
4	joe	que	her
5	que	her	joe
6	que	joe	her

正しい対応の可能性 6 通り

あなた「3 個のものを並べる順列の数ですから  $3! = 3 \times 2 \times 1 = 6$  通りあるということですか.....」

依頼者「はい。もしも復元途中のファイルが 159 265 314 = her joe que という 1 行のみからなるとするなら、正しい対応は上の 6 通りのいずれかであることはわかりますが、これ以上は何もいえません。ですから、159 265 314 = her joe que 以上に対応を復元することはできません」

あなた「はい」

依頼者「それではこちらのファイル sample.txt を見てください。このファイルは 4 行からなっています」

```
159 265 314 = her joe que
159 323 358 979 = asuka joe mari shinji
265 314 846 = her que rei
265 979 = her shinji
```

ファイル sample.txt

依頼者「これを見てよく考えると、265 = her であることがわかるはずです。265 という id は他のどの name にも対応づけることはできませんし、her という name は他のどの id にも対応づけることはできません」

あなた「なるほど、そうですね」

依頼者「さまざまな組み合わせを試してみればわかるのですが、結局、ファイル sample.txt から最大限に復元したファイルとして、次のファイル sample\_answer.txt が得られます。323 358 = asuka mari は、これ以上復元することはできません」

159 = joe  
265 = her  
314 = que  
323 358 = asuka mari  
846 = rei  
979 = shinji

ファイル sample.txt から最大限に復元したファイル sample\_answer.txt

あなた「なるほど、わかりました」

依頼者「さて、もちろんここまではご説明のための例に過ぎません。ここからがお仕事です。あなたに復元途中のファイルとして shufflers.txt をお渡しします。そこから可能な限り多くの “id = name” という対応を復元した answer.txt を作っていただきたいのです」

あなた「なるほど。お話はよくわかりました。ところで仕事の評価はどうなりますか」

依頼者「お仕事は復元率で評価させていただきます。つまり、

$$\text{復元率} = \frac{\text{《あなたが answer.txt で復元した “id = name” という対応の個数》}}{\text{《shufflers.txt に含まれているユニークな id の個数》}}$$

が 100 % に近ければ近い方が高い評価となります。ただし、answer.txt では、sample\_answer.txt における 323 358 = asuka mari のような、完全には復元できなかったものも途中経過という意味で含めておいてください」

あなた「わかりました。そうだ、もう一つだけ質問があります。shufflers.txt から復元率 100 % の answer.txt は作れるのでしょうか」

依頼者「それは不明です」

あなた「そうですか……ではファイル shufflers.txt をいただけますか」

依頼者「こちらです」

ミッションはこのようにして始まりました……。