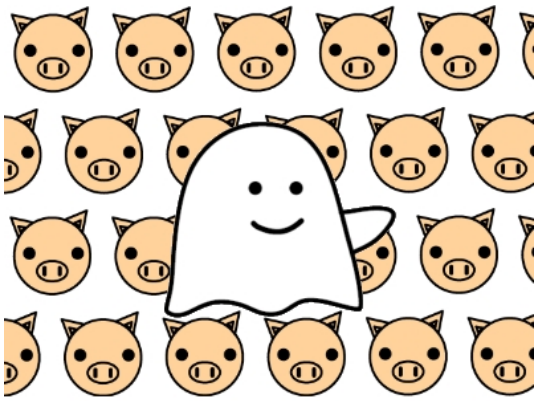
ピッグデータ問題

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

2013 年 4 月



注意： 本文書はピッグデータ問題についての《技術ドキュメント》です。

1　概要

この《技術ドキュメント》には、

1. 《ピッグデータ》と呼ばれる大量のデータを生成する方法
2. 《ピッグデータ》から《シグネチャ》と呼ばれる数を得る方法

が記述されています。この記述をもとにして、

A. 《シグネチャ》を得る（正確な内容はこの文書の末尾に改めて記載します）
B. 《シグネチャ》を得る際に自分が考えた内容を簡潔に説明した《技術メモ》を書く

の二つに解答するのが、あなたのミッションです。

2 《ピッグデータ》

《ピッグデータ》とは、大量の**データ**が格納されている、読み取り専用の仮想的な配列です（これはこの問題用に作った造語です）。ここでいうデータとは 0 以上の整数をいいます。

《ピッグデータ》に格納されている個々のデータは、すべて通常の配列同様に**インデクス**で管理されています。インデクスも 0 以上の整数です。

インデクスからデータを得るための関数を getdata と呼びます。getdata の関数仕様を以下に示します。

関数 getdata(index)

- 引数 index は 0 以上の整数である。
- 以下では index を 10 で割って小数以下を切り捨てた整数値を q と呼び、index を 10 で割った余りを r と呼ぶ。すなわち index $= 10q + r$ である（$0 \leqq r < 10$）。
- getdata は以下の手順で戻り値を計算する。
  - q を十進表記し、ASCII コードを使って文字列 s を作る。
  - 文字列 s には、C 言語にあるような終端の'\0' は含まれない。
  - 文字列 s をバイト列と見なし、メッセージダイジェストアルゴリズム SHA-1 を用いて 20 バイトのダイジェスト値 d[0]...d[19] を求める。
  - 2 バイト d[$2r$] と d[$2r+1$] を符号なし 16 ビット整数 data と見なす
    （ここで、d[$2r$] が上位 8 ビットであり、d[$2r+1$] が下位 8 ビットとする）。
  - このようにして得た data を getdata(index) の戻り値とする。

注意： 本来は《ピッグデータ》を二次記憶デバイスに格納してあなたに渡したいところですが、それはいささか難しいので、関数 getdata の仕様を渡すことで、《ピッグデータ》をあなたに渡したと見なしてください。

2.1 例 1

getdata(0) の値を求めます。この場合 $q = 0, r = 0$ になります。

文字列 “0” の SHA-1 によるダイジェスト値は、16 進数表記で b6 58 9f c6 ab 0d c8 2c f1 20 99 d1 c2 d4 0a b9 94 e8 41 0c ですから、d[$2r$] は b6 で、d[$2r+1$] は 58 です。

b6 は 10 進表記では 182 で、58 は 10 進表記では 88 です。

よって、getdata(0) の値は、$182 \times 256 + 88 = 46680$ となります。

2.2 例 2

getdata(123456) の値を求めます。この場合 $q = 12345, r = 6$ になります。

文字列 “12345” の SHA-1 によるダイジェスト値は、16 進数表記で 8c b2 23 7d 06 79 ca 88 db 64 64 ea c6 0d a9 63 45 51 39 64 ですから、d[$2r$] は c6 で、d[$2r+1$] は 0d です。

c6 は 10 進表記では 198 になり、0d は 10 進表記では 13 になります。

よって、getdata(123456) の値は、$198 \times 256 + 13 = 50701$ となります。

3 《シグニチャ》

《シグニチャ》とは、以下の仕様を満たす関数 `getsign(count, skips)` によって《ピッグデータ》から得られる 0 以上の整数です。

関数 `getsign(count, skips)`

- 引数 `count` は 1 以上の整数である。
- 引数 `skips` は 1 以上の整数である。
- `getsign` は以下の手順で戻り値を計算する。
  - 変数 `index` を 0 以上 `count` 未満の範囲で 1 ずつ増加させて `getdata(index)` を呼ぶ。
  - 得られた `count` 個のデータを**昇順でソートすると得られる数列**を `sorted[0]`, ..., `sorted[count − 1]` と呼ぶ。昇順なので、以下が成り立つ。

$$\texttt{sorted[0]} \leqq \texttt{sorted[1]} \leqq \cdots \leqq \texttt{sorted[count} - 1\texttt{]}$$

  - k を、$0 \leqq k <$ `count` かつ `skips` の倍数の範囲で動かしたときの `sorted[`k`]` の総和を `getsign(count, skips)` の戻り値とする
    （すなわち、k を $0, 1\texttt{skips}, 2\texttt{skips}, 3\texttt{skips}, \ldots$ のように動かして（ただし `count` 未満）、`sorted[`k`]` の総和を取るということである）。

3.1 例

`getsign(100, 10)` の値を求めます。《ピッグデータ》から得た `getdata(0)` から `getdata(99)` の値は、サンプルピッグデータ （ファイル `sample_pigdata.txt`）に書かれています。これをデータの昇順にソートすると、ファイル `sample_sorted.txt` のようになります。`index` の値が $0, 10, 20, 30, 40, 50, 60, 70, 80, 90$ の値の総和を取ると、

$$1194 + 6443 + 13674 + 20062 + 25722 + 31788 + 38120 + 44070 + 49631 + 53363 = 284067$$

よって、`getsign(100, 10)` の値は 284067 です。

4 あなたへの問題

A. 《シグネチャ》を得る
   ただし、`count` = 107374182400, `skips` = 16777216 とします。
   すなわち、`getsign(107374182400, 16777216)` の値を求めてください。
B. 《シグネチャ》を得る際に自分が考えた内容を簡潔に説明した《技術メモ》を書く

解答にあたっては、サンプル解答ファイル `answer.txt` も参照してください。