サイエンス・パートナーシップ・プロジェクト

暗号の基礎から実用まで

第2部：Python を使った数値計算

高知大学 理学部 応用理学科
塩 田 研 一
mail: shiota@is.kochi-u.ac.jp
http://lupus.is.kochi-u.ac.jp/~shiota/

2009年9月3日

現代の暗号で扱う数字：

10進数で何百桁

プログラム言語 Python なら簡単に

暗号処理の基礎となる計算

サンプルプログラムで体験を

はじめに

コンピュータはどんな計算も一瞬？　→　**No**

- 当たり前に高速にできる計算
- 工夫をすれば高速にできる計算
- どんなに工夫しても高速にはならない計算

の3種類ある

目　次

- ビット
- 高速にできる計算
  - 四則演算
- 工夫すれば高速にできる計算
  - 最大公約数
  - てんびんクイズ
  - 素数判定
  - べき乗計算
- 工夫しても高速にできない計算
  - 素因数分解

ビット

- ビット：2進数で表したときの桁のこと
- $53 = (110101)_2$

  - 53 は 6 ビット
  - 53 のビット長は 6

- k ビットの10進数はおおよそ $0.3 \times k$ 桁

実　習

サンプルプログラム「10進数から2進数.py」で2進数への変換をしてみよう

1.USBメモリの Python25 を開けて
2.IDLE.bat をダブルクリック
3.File → Open → 02_Shiota
4.10進数から2進数.py を選択
5.ファンクションキー F5 で実行

10進数から2進数.py の使い方

10進数を2進数へ変換してくれる

- 3^{50} は 3**50 と入力する
- 123 * 456 + 789 のように計算式を入力にすることもできる

Q1　100万は何ビットですか？

Q2　10億は何ビットですか？

問　題

1. 30ビットの数＋20ビットの数＝約何ビット？
2. 30ビットの数－20ビットの数＝約何ビット？
3. 30ビットの数×20ビットの数＝約何ビット？
4. 30ビットの数÷20ビットの数＝約何ビット？
5. 30ビットの数の平方根　　＝約何ビット？

Q & A

ご質問、ご意見ありましたら

高速にできる計算

四則演算

- 加法
- 減法
- 乗法
- 除法

コンピュータの内部では2進数で計算

2進数の筆算

$$
\begin{array}{rr}
 & 1\,1\,0\,1\,0\,0\,1 \\
+ & \quad 1\,1\,0\,0\,1\,1 \\
\hline
 & 1\,0\,0\,1\,1\,1\,0\,0
\end{array}
\qquad
\begin{array}{rr}
 & 1\,1\,0\,1\,0\,0\,1 \\
- & \quad 1\,1\,0\,0\,1\,1 \\
\hline
 & \quad 1\,1\,0\,1\,1\,0
\end{array}
$$

$$
\begin{array}{rr}
 & \quad 1\,1\,1\,0\,1 \\
\times & \quad 1\,0\,1\,1 \\
\hline
 & \quad 1\,1\,1\,0\,1 \\
 & 1\,1\,1\,0\,1 \\
1\,1\,1\,0\,1 & \\
\hline
1\,0\,0\,1\,1\,1\,1\,1\,1
\end{array}
\qquad
\begin{array}{r}
\quad 1\,0\,1 \\
\hline
1\,0\,1\,)\,1\,1\,1\,0\,1 \\
\quad 1\,0\,1 \\
\hline
\quad 1\,0\,0\,1 \\
\quad 1\,0\,1 \\
\hline
\quad 1\,0\,0
\end{array}
$$

2進数の加法

各桁では

- 繰り上がり無しの場合

  0+0=0,　　0+1=1,

  1+0=1,　　1+1=10

- 繰り上がり有り（1）の場合

  1+0+0=1,　1+0+1=10,

  1+1+0=10,　1+1+1=11

⇒ ビット数に比例した計算時間

2進数の乗法

筆算の各段で足し算をするイメージ

⇒　筆算の面積に比例した計算時間

実　習

サンプルプログラム「四則演算.py」で
計算時間を計測してみよう

- 演算の番号を入力すると測定開始
- それぞれ1万回の計算時間を表示

Q ビット数が2倍になると計算時間は大
体何倍くらいになっているでしょうか？

Q & A

ご質問、ご意見ありましたら

工夫すれば高速にできる計算
その1：最大公約数

最大公約数はこんなところにも …

てんびんクイズ

- 大きなてんびんと
- a グラムと b グラムの分銅たくさん

があるとき、

Q　測れる一番小さな重さは何グラム？

- 3 グラムと 5 グラム

  ⇒ $3\times2=5\times1+1$ で 1 グラム

- 5 グラムと 7 グラム

  ⇒ $5\times3=7\times2+1$ で 1 グラム

- 7 グラムと 11 グラム

  ⇒ $7\times3+1=11\times2$ で 1 グラム

- 6 グラムと 10 グラム

  ⇒ $6\times2=10\times1+2$ で 2 グラム

- 15 グラムと 21 グラム

  ⇒ $15\times3=21\times2+3$ で 3 グラム

- 28 グラムと 44 グラム

  ⇒ $28\times3+4=44\times2$ で 4 グラム

てんびんクイズの答え

- a と b が互いに素

  ⇒ 1 グラム

- a と b の最大公約数が d

  ⇒ d グラム

最大公約数の計算方法 1

素因数分解して共通する項を拾う

$$48 = 2^4 \times 3, \quad 108 = 2^2 \times 3^3$$

48 と 108 の最大公約数は

$$2^2 \times 3 = 12$$

最大公約数の計算方法 2

小さい方の数から 1 ずつ減らして割ってみる

108 ÷ 48 は余り 12,
108 ÷ 47 は余り 14,
108 ÷ 46 は余り 16,
:
108 ÷ 12 は余り 0,

48 と 108 の最大公約数は 12

最大公約数の計算方法 3

ユークリッドのアルゴリズム：

(a を b で割った余り) = c

a と b の最大公約数 $\gcd(a,b)$

= b と c の最大公約数 $\gcd(b,c)$

これより　$\gcd(108,48) = \gcd(48,12) = 12$

実　習

サンプルプログラム「最大公約数.py」で計算時間を計測してみよう

10ビット位から少しずつ大きくして …

Q & A

ご質問、ご意見ありましたら

定　理

てんびんクイズを式に書き直すと

a, b を与えたとき、

$$ax + by$$

の形で書ける最小の自然数は
最大公約数 $\gcd(a,b)$ である。

… x, y の絶対値が分銅の個数

特に

a, b を与えたとき、

$$\gcd(a,b) = ax + by$$

を満たす x, y が必ず存在する。

その x, y の求め方:

$x = 1$ から順番に探す？

時間が掛かり過ぎ

ユークリッドのアルゴリズムを改良

実　習

サンプルプログラム「てんびん.py」
で計算時間を計測してみよう

やはり10ビット位から少しずつ大きくして …

Q & A

ご質問、ご意見ありましたら

工夫すれば高速にできる計算
その２：素数判定

問　題

1. 137 は素数でしょうか？

   素数です。

2. 187 は素数でしょうか？

   素数ではありません。

   $187 = 11 \times 17$

素朴な素数判定法

n が素数

$\Leftrightarrow$ n の平方根以下の約数は 1 しかない

理由: a が n の約数

$\Rightarrow$ $b = n/a$ も n の約数

a, b のどちらかは n の平方根以下

高級な素数判定法

さっきの素数判定法:

約数の候補が沢山あり過ぎ

実用には確率的素数判定法を

実　習

サンプルプログラム「素数生成.py」
で計算時間を計測してみよう

30ビット位から少しずつ大きくして …

Q & A

ご質問、ご意見ありましたら

工夫すれば高速にできる計算
その3：べき乗計算

問　題

x の数値が与えられたとき、

Q1　x^{64} は何回の掛け算で計算できるでしょうか？

Q2　x^{83} は何回の掛け算で計算できるでしょうか？

Q1 の答え

$$x^2 = x \times x$$
$$x^4 = x^2 \times x^2$$
$$x^8 = x^4 \times x^4$$
$$x^{16} = x^8 \times x^8$$
$$x^{32} = x^{16} \times x^{16}$$
$$x^{64} = x^{32} \times x^{32}$$

で 6 回。

Q2 の答え

x^{64} を計算するまでに 6 回、

$$x^{83} = x^{64} \times x^{16} \times x^{2} \times x$$

で更に 3 回、合計 9 回。

これを「反復2乗法」という。

実　習

サンプルプログラム「べき乗.py」
で計算時間を計測してみよう

暗号で使う数字は 10^{300} 位
… 1 回ずつ掛けたら宇宙が終わるよ

Q & A

ご質問、ご意見ありましたら

工夫しても高速にできない計算

素因数分解

素朴な素因数分解法

さっきと同じく、

n が合成数

⇒　n の平方根以下の約数がある

しかし！　候補が多過ぎ

高級な素因数分解法

が色々ある:

- 連分数法
- 二次ふるい法
- 複数多項式二次ふるい法
- 楕円曲線法　　etc.

しかし！　数が大きくなるとやはり時間が掛かり過ぎる

実　習

サンプルプログラム

- 「素朴な素因数分解.py」
- 「高級な素因数分解.py」

で計算時間を計測してみよう

20ビット×20ビット位から少しずつ大きくして …

Q & A

ご質問、ご意見ありましたら

まとめ

- 当たり前に高速にできる計算
  - 四則演算（でも一瞬ではない）
- 工夫をすれば高速にできる計算
  - 最大公約数
  - てんびんクイズ
  - 素数判定
  - べき乗
- どんなに工夫しても高速にはならない計算
  - 素因数分解

次回は

- 当たり前に高速にできる計算
- 工夫をすれば高速にできる計算
- どんなに工夫しても高速にはならない計算

を組み合わせて暗号を作りましょう。