

# アルゴリズム論

2017年5月29日

樋口文人

# 目次

- 解説
- 集合
- ビット配列
- 課題

# 集合

# 辞書

- キーと値のペア
- 日常的には...
  - 見出し語とその意味／解説のペアを
  - 見出し語に基づく順番に並べたもの
- 使用法
  - 見出し語を探し, その意味を知る
  - 見出し語の数と種類は決まっている
    - 見出し語が増減したり変化したりはしない

# 集合

- キーが存在するかどうかの問題の場合
- 実現手段
  - 辞書で存在するかどうかを表す値を使う
    - ハッシュテーブル
  - キーの数が少ない, なるべく資源を使わない
    - ビット配列

# 1つの集合に対する操作

- 要素を加える
- 要素を取り除く
- 要素を探す
  - 任意の要素の有無を知る

# 複数の集合に対する操作

- 関係

- 部分集合:  $A \subseteq B, A \subset B$

- 演算

- 和集合:  $A \cup B$

- 積集合:  $A \cap B$

- 差集合:  $A - B = \{x \mid x \in A \wedge \neg(x \in B)\}$

# ビット配列

- 集合をつくる各要素の有無を1ビットで表し,  
要素間の区別をビットの位置で表したもの
  - CPU固有のメモリへのアクセス単位(ワード)の大きさに依存
  - 大きな集合が必要な場合には複数ワードを使用



# ビット配列の例

- 1ワードが32ビットとして
  - ある月の休日(1年分では12ワード)
  - 01000000100000010000001000000100
- ファイルのアクセス権
  - rwxrw-rw- ... 766 ... 111110110
- 血液型
  - A因子-B因子-Rh因子
    - A+ ... 101

# ビット演算

- ビット毎の論理和: 和集合
  - 01011010 :  $A = \{ b, d, e, g \}$
  - 10101010 :  $B = \{ a, c, e, g \}$
  - 11111010 :  $A \vee B = \{ a, b, c, d, e, g \}$
- ビット毎の論理積: 積集合
  - 01011010 :  $A = \{ b, d, e, g \}$
  - 10101010 :  $B = \{ a, c, e, g \}$
  - 00001010 :  $A \wedge B = \{ e, g \}$

# ビット演算

- 差

- 01011010 :  $A = \{b, d, e, g\}$

- 10101010 :  $B = \{a, c, e, g\}$

- 01010000 :  $A - B = \{b, d\}$

- 11110000 :  $A \text{ XOR } B = \{a, b, c, d\}$

- 01011010 :  $A = \{b, d, e, g\}$

- 01010000 :  $A \wedge (A \text{ XOR } B) = \{b, d\} = A - B$

# クイズ

- ビット配列で集合を表すときどのような操作・演算により下記を実現できるか？
- 集合  $A, B$  について
  1.  $A = B$
  2.  $A \subset B$
- 要素  $x$  について
  1.  $x$  を  $A$  に加える
  2.  $x$  を  $A$  から取り除く
  3.  $x$  が  $A$  の中にあるかどうか探す

# 部分集合

- 同等

- 01011010 :  $A = \{b, d, e, g\}$
- 01011010 :  $B = \{b, d, e, g\}$
- 00000000 :  $A \text{ XOR } B = \{\} = 0 \therefore A = B$

- 包含関係

- 01011010 :  $A = \{b, d, e, g\}$
- 01011011 :  $C = \{b, d, e, g, h\}$
- 00000001 :  $A \text{ XOR } C = \{h\} \neq 0$
- 00000000 :  $A \wedge (A \text{ XOR } C) = 0 \therefore A \subset C$

# 加える, 取り除く, 探す

- 加える
  - $01011010 : A = \{ b, d, e, g \}$
  - $00100000 : c = \{ c \}$
  - $01111010 : A \vee c = \{ b, c, d, e, g \}$
- 取り除く
  - $01011010 : A = \{ b, d, e, g \}$
  - $00001000 : e = \{ e \}$
  - $11110111 : \neg e$
  - $01010010 : A \wedge \neg e = \{ b, d, g \}$
- 探す
  - $01011010 : A = \{ b, d, e, g \}$
  - $00010000 : d = \{ d \}$
  - $00010000 : A \wedge d = \{ d \} > 0$

# クイズ

- 今, 16ビットのビット配列である集合が表現されている. 何個の要素を含んでいるか求める手順を考えよ.

## ーヒント

- 4ビットの配列 0110 は第2要素と第3要素を含むことを表現しています
- 1ビットのビット配列で表された集合が4個並んでいると考えると, 0110 は要素0個を含む集合が2つ, 要素1個を含む集合が2つあることになります

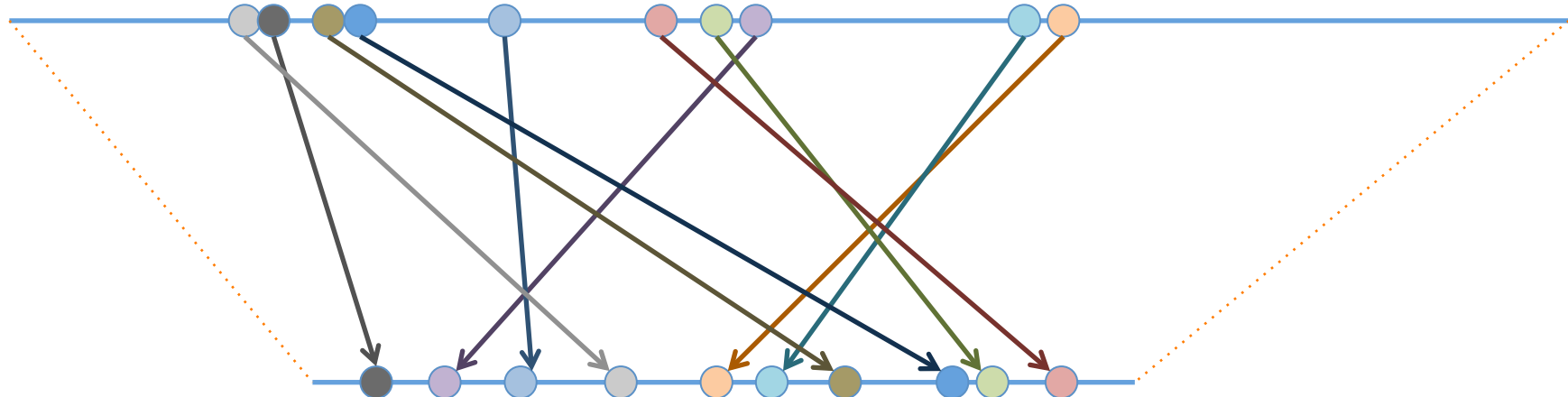
# 時間があれば...

- 8 bit 整数,  $a = 132$ ,  $b = 237$  がある。下記の順にビット演算をした後の  $a$ ,  $b$  の値は何か
  1.  $a = (a \text{ XOR } b)$
  2.  $b = (a \text{ XOR } b)$
  3.  $a = (a \text{ XOR } b)$
- 以下のビットパターンを持つ数値がある。下記のそれぞれの場合の大小関係を示せ
  - $a = 0111011001000101$
  - $b = 1100101000110110$
  - 1. 符号なし整数
  - 2. 符号+絶対値
  - 3. IEEE754に準拠した16bit浮動小数点数(指数部5bit、仮数部10bit)



# キーの変換写像(ハッシュ関数)

キーの空間



ハッシュテーブル(配列)の添え字空間

# 宿題: ex06

- 以下のファイルについてmd5 と sha256 のハッシュ値(チェックサム)を求め報告してください
  1. Oh-o!Meiji クラスウェブ 第1回授業の講義資料(pdf)ファイル
  2. これまでに宿題で提出したプログラム(プレーンテキスト)
    - 1文字程度の変更の前後を比較する
      1. 改行・スペースを加える／取る
      2. コメントなどの修正, 誤字の訂正, など
- 以上についての感想をまとめる

# ツール

- Mac OS X
  - ターミナル
    - md5, shasum
- Linux (Ubuntu)
  - ターミナル(端末)
    - md5sum, shasum, sha256sum
- Windows
  - PowerShell (v4以降)
    - Get-FileHash
  - それ以前
    - FCIV: md5 のみ, sha256 は未対応
  - その他, フリーウェア
    - MD5&SHA Checksum Utility: [http://download.cnet.com/MD5-SHA-Checksum\\_Utility/3000-2092\\_4-10911445.html](http://download.cnet.com/MD5-SHA-Checksum_Utility/3000-2092_4-10911445.html)

# ツール

## Mac OS X

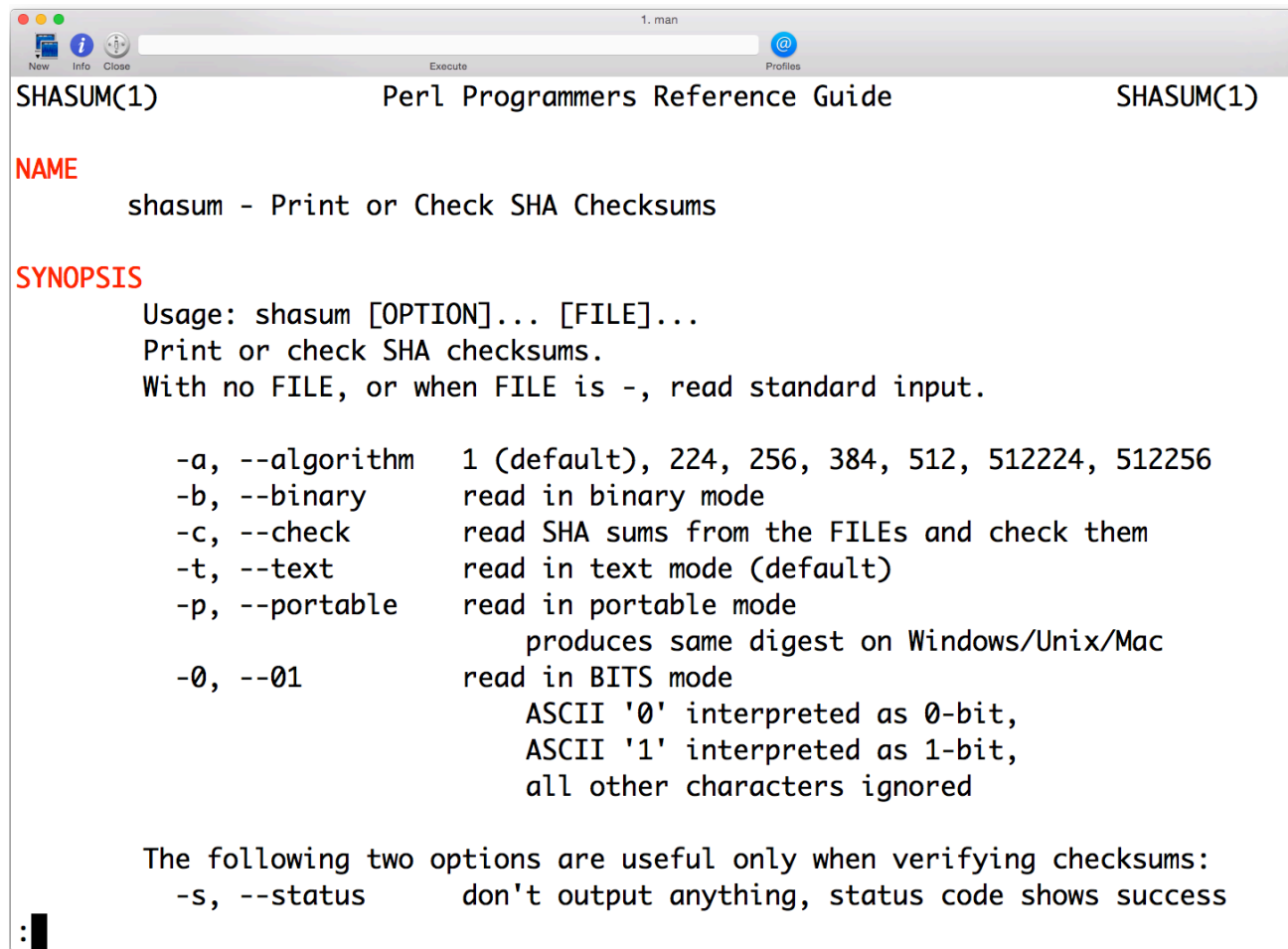


```
yuzu:~ wenren$ which md5
/sbin/md5
yuzu:~ wenren$ which shasum
/usr/bin/shasum
yuzu:~ wenren$
```

## Windows

A screenshot of the Download.com website for the 'MD5 &amp; SHA Checksum Utility'. The page features a green 'Get It Here' button at the top. Below it, there's a section for 'Quick Specs' with details like Version 2.1, 311,679 total downloads, and a price of 'Free'. The 'Editors' Review' section includes a description of cryptographic hash functions and a 'read more' link. The 'Publisher's Description' section mentions it's a standalone freeware tool. The page also includes social media links, a star rating, and several advertisements on the right side.

# man shasum



```
SHASUM(1)                                Perl Programmers Reference Guide                                SHASUM(1)

NAME
    shasum - Print or Check SHA Checksums

SYNOPSIS
    Usage: shasum [OPTION]... [FILE]...
    Print or check SHA checksums.
    With no FILE, or when FILE is -, read standard input.

    -a, --algorithm 1 (default), 224, 256, 384, 512, 512224, 512256
    -b, --binary    read in binary mode
    -c, --check     read SHA sums from the FILEs and check them
    -t, --text      read in text mode (default)
    -p, --portable  read in portable mode
                    produces same digest on Windows/Unix/Mac
    -0, --01        read in BITS mode
                    ASCII '0' interpreted as 0-bit,
                    ASCII '1' interpreted as 1-bit,
                    all other characters ignored

    The following two options are useful only when verifying checksums:
    -s, --status    don't output anything, status code shows success
```

# 補足

- md5やshaはデジタル署名や認証などセキュリティのために使用されています
- 十分なセキュリティが確保できなくなった技術は、より安全な技術に移行するために使われなくなります
  - md5 や sha-1 に代わって sha-2 (sha224, sha256, sha384, sha512, etc.) が使われるようになりつつあります

# 提出についての注意

- 今回はレポートの提出です
  - Microsoft Word形式のファイルで提出
    - Pages からは ファイル＞書き出す＞Word
    - リッチテキストフォーマット(rtf)ファイルでもよい
  - 冒頭に氏名、学科、学年、クラス、番号等を記すこと
  - ハッシュ値の計算に使用したProcessingのプログラムの番号 (eg. ex01, ex02, ex03, ..., ex05) を記すこと
  - どのような変更かも知れずに
- Oh-o!Meijiから提出(次回の授業開始までに)

# 連絡先

樋口文人

wenren@meiji.ac.jp