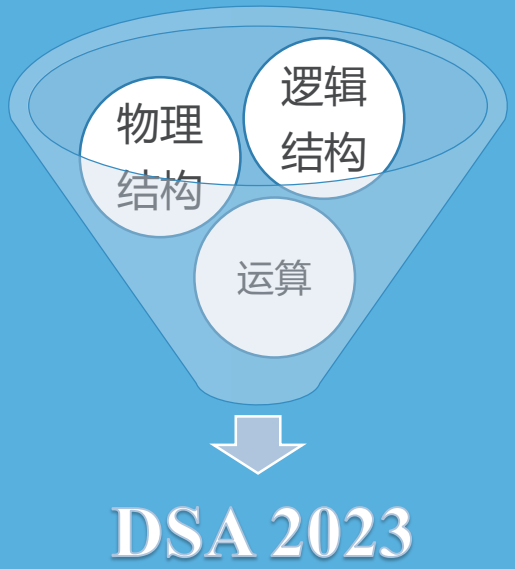


数据结构与算法（Python）-05+/MD5

刘云淮 Yunhuai.liu@pku.edu.cn

http://www.yunhuai.net/DSA2023/CoursePage/DSA2023.html

北京大学计算机学院

MD5 Specification

- Works on 512 bit blocks of the message
- Produces a 128 bit hash code

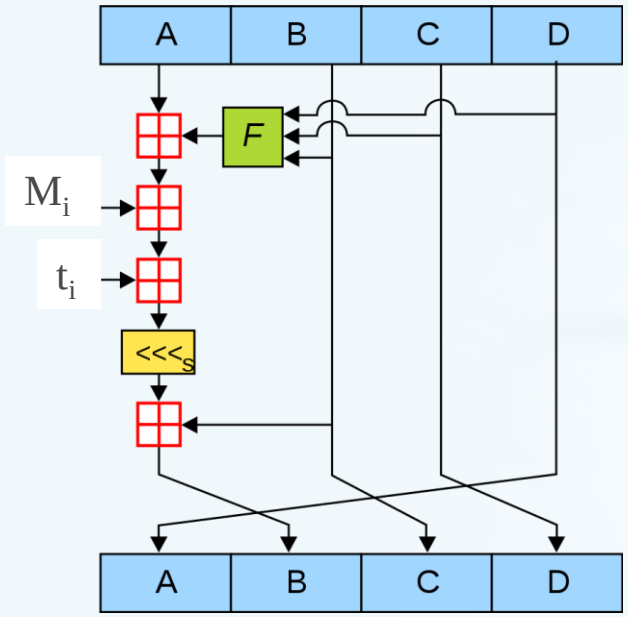
Message Preparation

- **Padding**
  The Message is padded to an exact multiple of 512-bit blocks
  1 is appended to message
  The remainder (less 64 bits) is filled with as many 0's as required
  The last 64 bits are used to represent the message length
- **Block subdivision**
  Subdivided to a number of 512-bit blocks

Basic Idea

- Start from a constant 128-bit state
- Divided into four 32-bit words, denoted as A, B, C, and D
- Each 512-bit block will be divided to 16 32-bit block
- Each 32-bit block will be used to modify this 128-bit state for 16 times
- Every 16 operations are called *one round*
- 4 rounds, with slightly different operations

Main Operation in Each Round



Block 1

Block 2

Block 3

Block 4

⋮

Block 16

Initialized Chaining Variables

A = 0x01234567
B = 0x89abcdef
C = 0xfedcba98
D = 0x76543210

Nonlinear Generating Functions

$$
\begin{aligned}
F(B,C,D) &= (B \wedge C) \vee (\neg B \wedge D) \\
G(B,C,D) &= (B \wedge D) \vee (C \wedge \neg D) \\
H(B,C,D) &= B \oplus C \oplus D \\
I(B,C,D) &= C \oplus (B \vee \neg D)
\end{aligned}
$$

$\mathrm{FF}(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + \mathrm{F}(b,c,d) + M_j + t_i) <<< s)$

$\mathrm{GG}(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + \mathrm{G}(b,c,d) + M_j + t_i) <<< s)$

$\mathrm{HH}(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + \mathrm{H}(b,c,d) + M_j + t_i) <<< s)$

$\mathrm{II}(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + \mathrm{I}(b,c,d) + M_j + t_i) <<< s)$

Basic Operations

$FF(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + F(b,c,d) + M_j + t_i) <<< s)$

$GG(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + G(b,c,d) + M_j + t_i) <<< s)$

$HH(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + H(b,c,d) + M_j + t_i) <<< s)$

$II(a,b,c,d,M_j,s,t_i)$ denotes $a = b + ((a + I(b,c,d) + M_j + t_i) <<< s)$

Some constants

M_j is the j^{th} sub-block of the message block.
For step i:
$t_i = 2^{32}$*abs(sin(i)) where i is measured in radians.

s is the number of bits to be shifted:
Round 1: [7, 12, 17, 22]
Round 2: [5, 9, 14, 20]
Round 3: [4, 11, 16, 23]
Round 4: [6, 10, 15, 21]

Round 1

FF (*a*, *b*, *c*, *d*, M_0, 7, 0xd76aa478)
FF (*d*, *a*, *b*, *c*, M_1, 12, 0xe8c7b756)
FF (*c*, *d*, *a*, *b*, M_2, 17, 0x242070db)
FF (*b*, *c*, *d*, *a*, M_3, 22, 0xc1bdceee)
FF (*a*, *b*, *c*, *d*, M_4, 7, 0xf57c0faf)
FF (*d*, *a*, *b*, *c*, M_5, 12, 0x4787c62a)
FF (*c*, *d*, *a*, *b*, M_6, 17, 0xa8304613)
FF (*b*, *c*, *d*, *a*, M_7, 22, 0xfd469501)
FF (*a*, *b*, *c*, *d*, M_8, 7, 0x698098d8)
FF (*d*, *a*, *b*, *c*, M_9, 12, 0x8b44f7af)
FF (*c*, *d*, *a*, *b*, M_{10}, 17, 0xffff5bb1)
FF (*b*, *c*, *d*, *a*, M_{11}, 22, 0x895cd7be)
FF (*a*, *b*, *c*, *d*, M_{12}, 7, 0x6b901122)
FF (*d*, *a*, *b*, *c*, M_{13}, 12, 0xfd987193)
FF (*c*, *d*, *a*, *b*, M_{14}, 17, 0xa679438e)
FF (*b*, *c*, *d*, *a*, M_{15}, 22, 0x49b40821)

Round 2

GG $(a, b, c, d, M_1, 5, 0xf61e2562)$
GG $(d, a, b, c, M_6, 9, 0xc040b340)$
GG $(c, d, a, b, M_{11}, 14, 0x265e5a51)$
GG $(b, c, d, a, M_0, 20, 0xe9b6c7aa)$
GG $(a, b, c, d, M_5, 5, 0xd62f105d)$
GG $(d, a, b, c, M_{10}, 9, 0x02441453)$
GG $(c, d, a, b, M_{15}, 14, 0xd8a1e681)$
GG $(b, c, d, a, M_4, 20, 0xe7d3fbc8)$
GG $(a, b, c, d, M_9, 5, 0x21e1cde6)$
GG $(d, a, b, c, M_{14}, 9, 0xc33707d6)$
GG $(c, d, a, b, M_3, 14, 0xf4d50d87)$
GG $(b, c, d, a, M_8, 20, 0x455a14ed)$
GG $(a, b, c, d, M_{13}, 5, 0xa9e3e905)$
GG $(d, a, b, c, M_2, 9, 0xfcefa3f8)$
GG $(c, d, a, b, M_7, 14, 0x676f02d9)$
GG $(b, c, d, a, M_{12}, 20, 0x8d2a4c8a)$

Round 3

HH (*a*, *b*, *c*, *d*, M_5, 4, 0xfffa3942)
HH (*d*, *a*, *b*, *c*, M_8, 11, 0x8771f681)
HH (*c*, *d*, *a*, *b*, M_{11}, 16, 0x6d9d6122)
HH (*b*, *c*, *d*, *a*, M_{14}, 23, 0xfde5380c)
HH (*a*, *b*, *c*, *d*, M_1, 4, 0xa4beea44)
HH (*d*, *a*, *b*, *c*, M_4, 11, 0x4bdecfa9)
HH (*c*, *d*, *a*, *b*, M_7, 16, 0xf6bb4b60)
HH (*b*, *c*, *d*, *a*, M_{10}, 23, 0xbebfbc70)
HH (*a*, *b*, *c*, *d*, M_{13}, 4, 0x289b7ec6)
HH (*d*, *a*, *b*, *c*, M_0, 11, 0xeaa127fa)
HH (*c*, *d*, *a*, *b*, M_3, 16, 0xd4ef3085)
HH (*b*, *c*, *d*, *a*, M_6, 23, 0x04881d05)
HH (*a*, *b*, *c*, *d*, M_9, 4, 0xd9d4d039)
HH (*d*, *a*, *b*, *c*, M_{12}, 11, 0xe6db99e5)
HH (*c*, *d*, *a*, *b*, M_{15}, 16, 0x1fa27cf8)
HH (*b*, *c*, *d*, *a*, M_2, 23, 0xc4ac5665)

Round 4

II (*a*, *b*, *c*, *d*, M_0, 6, 0xf4292244)
II (*d*, *a*, *b*, *c*, M_7, 10, 0x432aff97)
II (*c*, *d*, *a*, *b*, M_{14}, 15, 0xab9423a7)
II (*b*, *c*, *d*, *a*, M_5, 21, 0xfc93a039)
II (*a*, *b*, *c*, *d*, M_{12}, 6, 0x655b59c3)
II (*d*, *a*, *b*, *c*, M_3, 10, 0x8f0ccc92)
II (*c*, *d*, *a*, *b*, M_{10}, 15, 0xffeff47d)
II (*b*, *c*, *d*, *a*, M_1, 21, 0x85845dd1)
II (*a*, *b*, *c*, *d*, M_8, 6, 0x6fa87e4f)
II (*d*, *a*, *b*, *c*, M_{15}, 10, 0xfe2ce6e0)
II (*c*, *d*, *a*, *b*, M_6, 15, 0xa3014314)
II (*b*, *c*, *d*, *a*, M_{13}, 21, 0x4e0811a1)
II (*a*, *b*, *c*, *d*, M_4, 6, 0xf7537e82)
II (*d*, *a*, *b*, *c*, M_{11}, 10, 0xbd3af235)
II (*c*, *d*, *a*, *b*, M_2, 15, 0x2ad7d2bb)
II (*b*, *c*, *d*, *a*, M_9, 21, 0xeb86d391)