数据结构与算法（Python版）

完美散列函数

陈斌 北京大学 gischen@pku.edu.cn

完美散列函数

- 给定一组数据项，如果一个散列函数能把每个数据项映射到不同的槽中，那么这个散列函数就可以称为“**完美散列函数**”

  对于固定的一组数据，总是能想办法设计出完美散列函数

- 但如果数据项经常性的变动，很难有一个系统性的方法来设计对应的完美散列函数

  当然，冲突也不是致命性的错误，我们会有办法处理的。

完美散列函数

- 获得完美散列函数的一种方法是扩大散列表的容量，大到**所有可能出现**的数据项都能够占据不同的槽
- 但这种方法对于可能数据项范围过大的情况并不实用

  假如我们要保存手机号（11位数字），完美散列函数得要求散列表具有百亿个槽！会浪费太多存储空间
- 退而求其次，好的散列函数需要具备特性

  冲突最少（近似完美）、计算难度低（额外开销小）、充分分散数据项（节约空间）

完美散列函数的更多用途

- 除了用于在散列表中安排数据项的存储位置，散列技术还用在信息处理的很多领域
- 由于完美散列函数能够对任何不同的数据生成不同的散列值，如果把散列值当作数据的“指纹”或者“摘要”，这种特性被广泛应用在数据的一致性校验上

  由任意长度的数据生成长度固定的“指纹”，还要求具备唯一性，这在数学上是无法做到的，但设计巧妙的“准完美”散列函数却能在实用范围内做到这一点。

完美散列函数的更多用途

- 作为一致性校验的数据“指纹”函数需要具备如下的特性

  压缩性：任意长度的数据，得到的“指纹”长度是固定的；

  易计算性：从原数据计算“指纹”很容易；（从指纹计算原数据是不可能的）；

  抗修改性：对原数据的微小变动，都会引起“指纹”的大改变；

  抗冲突性：已知原数据和“指纹”，要找到相同指纹的数据（伪造）是非常困难的

散列函数MD5/SHA

- 最著名的近似完美散列函数是MD5和SHA系列函数
- MD5（Message Digest）将任何长度的数据变换为固定长为128位（16字节）的“摘要”

  128位二进制已经是一个极为巨大的数字空间：

  据说是地球沙粒的数量

散列函数MD5/SHA

- SHA（Secure Hash Algorithm）是另一组散列函数
  - SHA-0/SHA-1输出散列值160位（20字节），
  - SHA-256/SHA-224分别输出256位、224位，
  - SHA-512/SHA-384分别输出512位和384位
- 160位二进制相当于10的48次方，地球上水分子数量估计是47次方
- 256位二进制相当于10的77方，已知宇宙所有基本粒子大约是72～87次方

散列函数MD5/SHA

- 虽然近年发现MD5/SHA-0/SHA-1三种散列函数
- 能够以极特殊的情况来构造个别碰撞（散列冲突)
- 但在实用中从未有实际的威胁。
- 关于数量级的知识：
  http://zh.wikipedia.org/wiki/%E6%95%B0%E9%87%8F%E7%BA%A7_(%E6%95%B0)

Python的散列函数库hashlib

- Python自带MD5和SHA系列的散列函数库：hashlib

  包括了md5 / sha1 / sha224 / sha256 / sha384 / sha512等6种散列函数

```
>>> import hashlib
>>> hashlib.md5("hello world!").hexdigest()
'fc3ff98e8c6a0d3087d515c0473f8677'
>>> hashlib.sha1("hello world!").hexdigest()
'430ce34d020724ed75a196dfc2ad67c77772d169'
```

Python的散列函数库hashlib

- 除了对单个字符串进行散列计算之外，
- 还可以用update方法来对任意长的数据分部分来计算，
- 这样不管多大的数据都不会有内存不足的问题。

```
>>> import hashlib
>>> m= hashlib.md5()
>>> m.update("hello world!")
>>> m.update("this is part #2")
>>> m.update("this is part #3")
>>> m.hexdigest()
'a12edc8332947a3e02e5668c6484b93a'
>>>
```

完美散列函数用于数据一致性校验

- 数据文件一致性判断
- 为每个文件计算其散列值，仅对比其散列值即可得知是否文件内容相同；
- 用于网络文件下载完整性校验；
- 用于文件分享系统：网盘中相同的文件（尤其是电影）可以无需存储多次。

完美散列函数用于数据一致性校验

- 加密形式保存密码
- 仅保存密码的散列值，用户输入密码后，计算散列值并比对；
- 无需保存密码的明文即可判断用户是否输入了正确的密码。

完美散列函数用于数据一致性校验

- **防文件篡改：原理同数据文件一致性判断**

  当然还有更多密码学机制来保护数据文件，

  防篡改，防抵赖，是电子商务的信息技术基础。

- **彩票投注应用**

  彩民下注前，机构将中奖的结果散列值公布，

  然后彩民投注，开奖后，彩民可以通过公布的结果和散列值对比，验证机构是否作弊。