
问题解答

两周内容小结：W07/08：查找排序

完美散列函数的应用

区块链和比特币相关问题

# 数据结构与算法（Python）-13/0410

陈斌 gischen@pku.edu.cn 北京大学地球与空间科学学院

# 目录



- 问题解答
- 两周内容小结

  W07/08：查找排序
- 关于H4作业

# 问题解答？



- 请大家继续发在Canvas讨论里。

# 本章目标



- 了解和实现顺序查找和二分法查找；
- 了解和实现选择排序、冒泡排序、归并排序、快速排序、插入排序和希尔排序；
- 了解用散列Hashing实现查找的技术；
- 了解抽象数据类型：映射Map；
- 采用散列实现抽象数据类型Map。

# W07/08：查找与排序



- 501 顺序查找算法及分析 9m41s
- 502 二分查找算法及分析 12m20s
- 503 冒泡和选择排序算法及分析 12m14s
- 504 插入排序算法及分析 7m06s
- 505 谢尔排序算法及分析 6m15s
- 506 归并排序算法及分析 9m13s
- 507 快速排序算法及分析 12m30s

# W07/08：查找与排序



- 508 什么是散列 7m21s
- 509 完美散列函数 15m02s
- 510 区块链技术 17m20s
- 511 散列函数设计 8m47s
- 512 冲突解决方案 11m59s
- 513 映射抽象数据类型及Python实现 14m58s
- 514 排序与查找小结 9m45s

# 查找过程的信息



- **存取（access）：由位置存储/获取值**
- **查找（search）：由值确定其存储位置**
- **数据存储位置与其值无关：无序表**
  只有相等或不相等：顺序查找
- **数据存储的相对位置与其值相关：有序表**
  如果还可以进行大小比较：按照大小排序后，二分查找
- **数据存储的绝对位置与其值相关：散列表**
  值和存储位置有函数关系：通过散列函数直接映射到位置

# 排序过程中的比较信息



## 一阶信息

ai，aj之间的大小比较，如ai<aj

冒泡、选择、插入、谢尔排序

每一轮都会将未排序的部分捋一遍，未排序部分每轮缩小一个数

时间复杂度在$n^2$级别

# 排序过程中的比较信息



## 二阶的隐含信息

ai，aj以及ai，ak之间大小的信息

如ai>aj，又有aj>ak，应该有ai>ak，ai和ak不需要再次比较

快速排序：中值就是aj，将把ai和ak分裂到不同子表，不会再次比较

归并排序：在合并过程中，比如ai属一个子表，aj/ak属另一个子表，也不会让ai和ak再次比较

未排序部分每轮缩小一半，时间复杂度在nlogn级别

# 具有先验信息的排序算法



- **基数排序**

  通过把整数表示为某个进制的符号表示，如十进制；

  我们已经预先知道了进制中数字符号之间的大小；

  基数队列之间是有固定的先后次序。

- **这样甚至不需要去直接比较，只需要按照每一位符号排到相应的队列里**

- **所以具有更好的排序性能，时间复杂度是$n\log_B N$级别，B是基数（如10），N是这组数中最大的数，就是最大的数有几位。**

# 完美散列函数的应用



## 作为校验码防止出错

文件F，和Hash(F)，如果F的内容有任何错误变为F'

那么Hash(F')就与Hash(F)有显著区别

## 作为防止篡改/抵赖的手段

电子借据doc，和Hash(doc)一起发送……

且慢！如果A和B都声称自己的那个doc是原件怎么办？

# 完美散列函数的应用



- **基于数学难题的公开密钥加密，用于电子签名**

  两个特别大的素数P1和P2，其乘积PU=P1*P2

  用PU加密的信息，仅能用P1（或P2）解密，反之亦然

  PU称为“公开密钥”—公钥—公诸于众

  P1（或P2）称为“私有密钥”—私钥—要藏好勿泄漏

# 电子签名防止篡改/抵赖



- A借了B的钱，写下一个电子借据doc
- A将Hash(doc)，用自己的私钥AP1加密为AP1(Hash(doc))
- 把doc和AP1(Hash(doc))一起发送给B
- 这样，B和所有人都可以检查doc是不是原文

  `Hash(doc) == APU(AP1(Hash(doc)))`

  （用私钥加密的信息只能用公钥解密，A的公钥APU是众所周知的）
- A也无法抵赖说，从来没写过借据doc

  因为AP1(Hash(doc))在B手里，没有私钥AP1的人是无法算出这个签名的

# 问题解答



- 请问能否解释一下开放地址的散列查找，成功和失败的平均查找次数和负载因子关系，那两个公式是怎么推出来的？
- 请问老师能否详细讲一下散列查找函数的几个时间复杂度是如何计算出来的，以及它们各自的适用条件？

## 散列算法分析

- 如果采用线性探测的开放定址法来解决冲突（λ在0~1之间）

  成功的查找，平均需要比对次数为：$\frac{1}{2}\left(1+\frac{1}{1-\lambda}\right)$

  不成功的查找，平均比对次数为：$\frac{1}{2}\left(1+\left(\frac{1}{1-\lambda}\right)^2\right)$

- 如果采用数据链来解决冲突（λ可以大于1）

  成功的查找，平均需要比对次数为：1+λ/2

  不成功的查找，平均比对次数为：λ

# 问题解答



- **区块链在每一个节点把所有数据保存一遍，这有什么好处？这是否是一种极其浪费存储空间的行为？是不是为了保存所有数据，每一个节点都应当是巨型服务器？**

去中心化，不会单点失效，不被任何实体控制
decentralized

但没有中心或权威的话，难题是信任，谁说了算？
……让“工作量证明”说了算

# 问题解答



- **请问能详细解释一下区块链和比特币的概念吗？比如比特币的交易，交易记录是如何实现的？**

  ……见视频课件；交易记录是全网同步，但要经过工作量证明的竞争才能挂到区块链里去，成为确定的账本。

- **为什么比特币具有价值？为什么会有人花钱买比特币？**

  ……一般等价物，能够进行匿名的可信交易，在网络上实现纸钞的功能

- **挖矿时的计算是对比特币的使用有用的计算，还是无用的计算，完全就是在浪费算力？**

  ……工作量证明，大部分计算是无效的，但这是匿名互信的唯一途径，

  ……同时不断有算力参与记账，也是维持电子货币运转体系所必须的

  ……bitcoin天才地把这两者结合在一起

# 问题解答



- **请问利用散列函数的方式去比较电影的重复度从而减少反复存储的过程中，散列函数的自变量是什么呢？**

  ……就是电影文件的所有bits

  ……只比较两个文件的hash值，如果相同，就一定是同一个文件

- **想到另外一种开放定址的方式：如果该数据的散列函数值被占用的话，则把现在占用槽数据移位，将这个数据放进去。这样的方式和原来的开放定址查找的复杂度可比较吗？**

  ……ssfd，这本质上跟原来一样

# 关于本周作业



- **必做：见Canvas页面和gis4g公告**

  慕课在线测验