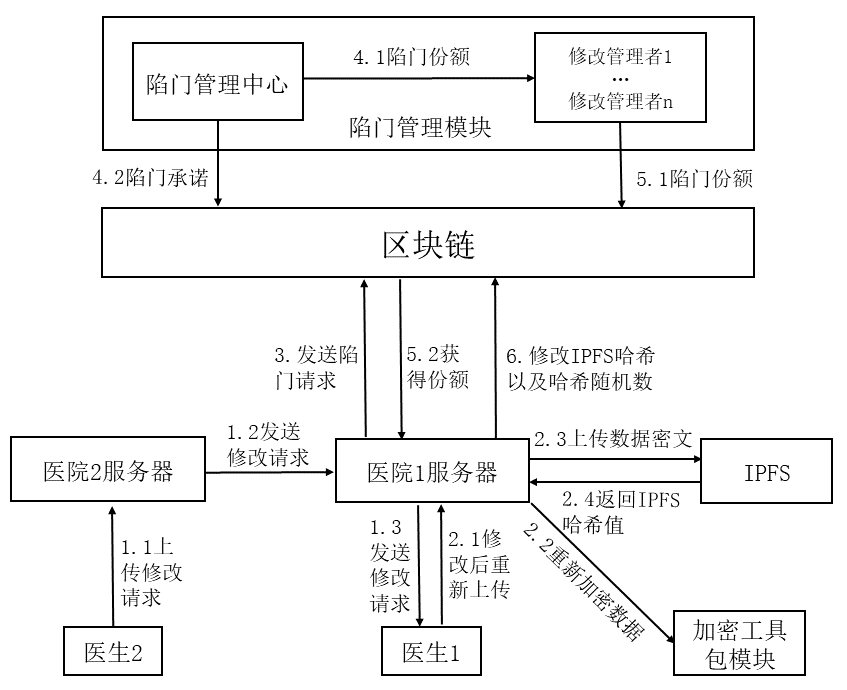
**说明书摘要**

//待填充

**摘要附图**

**// 待填充**



**权利要求书**

1. 一种基于B树和B+树与Bloom过滤器的云存储高效查找结构，其核心在于通过优化数据索引和检索机制，实现云存储系统中数据的快速定位与访问。该结构包括以下关键组成部分：云存储服务器、B树索引模块、B+树索引模块、Bloom过滤器模块以及用户接口；其中：

所述云存储服务器，作为中心节点，负责存储和管理海量数据，同时提供数据的写入和读取服务；

所述B树索引模块，采用多路平衡查找树结构，用于维护数据的索引信息，支持高效的数据插入、删除和查找操作。B树通过保持所有节点的平衡，确保数据的有序性，从而优化查找效率；

所述B+树索引模块，是B树的变种，它将所有的数据存储在叶子节点，并将叶子节点通过指针连接起来，形成链表结构。这种设计使得B+树在进行范围查询时更加高效，因为它可以快速地遍历叶子节点链表；

所述Bloom过滤器模块，用于快速判断某个元素是否可能存在于云存储中。Bloom过滤器是一种空间效率很高的概率型数据结构，它允许一定的误判率，但可以显著减少不必要的磁盘I/O操作，提高数据检索的速度；

所述用户接口，提供了一个友好的用户交互界面，用户可以通过该接口进行数据的上传、查询和下载操作，同时，用户接口会利用B树和B+树索引模块以及Bloom过滤器模块，以实现对数据的快速定位和检索。

通过这种结构，云存储系统能够在保证数据安全性和一致性的同时，显著提高数据检索的效率和性能。

在构建一种基于B树和B+树与Bloom过滤器的云存储高效查找结构中，我们采用了多层次的加密策略来确保数据的安全性。具体来说，我们设计了一个加密工具包，该工具包采用以下加密流程：

数据加密：首先，使用对称加密算法对云存储中的数据进行加密，生成数据密文CTCT。

密钥加密：然后，采用CP-ABE（Ciphertext-Policy Attribute-Based Encryption）算法对数据密钥进行加密，生成密钥密文ctct。

数据存储：最后，将加密后的数据密文CTCT和密钥密文ctct一起上传至分布式存储系统，如IPFS。

在这一过程中，云存储服务器扮演着关键角色，它不仅是数据存储的节点，也是用户（如医生）与云存储系统进行交互的中间节点。云存储服务器在数据管理过程中执行以下操作：

数据上传：将加密后的数据上传至分布式存储系统，并生成相应的元数据，这些元数据随后被上传至云存储系统中的B+树索引模块，以便于后续的数据检索。

数据查询：接收用户的查询请求（如文件标识、数据类型等），利用B树索引模块快速定位到符合条件的数据元数据，并调用Bloom过滤器进行初步筛选，以验证数据的存在性，最后返回有效的数据位置信息。

数据修改：当需要更新存储的数据时，云存储服务器会向B+树索引模块发布修改请求。一旦修改被批准，相关信息将在B+树索引模块中更新，并通过Bloom过滤器进行同步更新，确保数据的一致性和可检索性。

此外，为了进一步提高系统的安全性和数据的可修正性，我们引入了变色龙哈希函数。这是一种具有陷门特性的哈希函数，允许在保持数据完整性的同时对数据进行修正。陷门管理模块由一个可信的陷门管理中心和一组修改管理者组成，管理中心负责生成和分发陷门份额，而修改管理者则通过发布陷门份额来决定是否批准数据的修改请求。

用户，在这里特指需要访问云存储数据的医生，通过云存储服务器与系统进行交互，利用B树和B+树索引模块以及Bloom过滤器，实现对数据的快速定位和检索。

在所述的云存储系统中，总架构分为如下几个方面：

步骤 1:关键词和文档的准备

文档集合：系统里有多个文档（例如 m1, m2, m3, ..., mN-1, mN），每个文档都与一组关键词（Keywords）相关联。

关键词分类：每个文档的关键词集合可以被表示和分类，例如 K1, K2, K3, K4 等。

步骤 2:布隆过滤器的创建

布隆过滤器：对于每个文档的关键词集合，系统会创建一个布隆过滤器（Bloom Filter），这是一种用于高效测试元素是否为集合成员的数据结构。

加密和访问策略：每个文档的布隆过滤器和访问策略一起被加密，确保只有拥有适当属性的用户才能进行搜索。

步骤 3:云服务器上的加密

加密文档：加密后的文档（例如 ct1, ct2, ct3, ...）被存储在云服务器上。

索引和加密：每个文档的加密索引也存储在云服务器上，以支持快速检索。

步骤 4:用户查询

用户关键词查询：用户提交一个包含关键词的查询（例如K1,K2,K3）。

生成令牌：系统为用户的查询生成一个令牌（Token），这通常包括对用户身份和关键词的加密表示。

步骤 5:搜索和检索

搜索：云服务器使用用户的令牌来搜索匹配的文档。

布隆过滤器匹配：通过比较用户查询的布隆过滤器与文档的加密布隆过滤器来执行搜索。

步骤 6:返回结果

结果返回：搜索结果（可能包括文档的加密版本）被返回给用户。

解密和访问：用户使用他们的私钥解密并访问检索到的文档。

步骤 7:可视化和流程

图示：FAKS的流程通常被可视化，如图4所示，其中包括文档、关键词、布隆过滤器、加密、搜索和结果返回的各个组成部分。

这个架构的目的是提供一个既能保护文档内容和用户隐私，又能允许有效搜索的方法。通过使用布隆过滤器和属性基加密，FAKS旨在解决云存储环境中的安全和性能挑战。

**说明书**

一种基于可编辑区块链和属性密码的云存储高效查找结构

**技术领域**

本发明属于云存储领域，具体涉及基于可编辑区块链和属性密码、B树与B+树的云存储高效查找结构。

**背景技术**

在传统的基于中心化的云存储系统中，尽管具有较高的存储访问性能，但单点故障仍是其面临的最大挑战之一。一旦中心服务器遭受破坏或黑客攻击，可能导致整个系统瘫痪甚至数据永久丢失。因此，探索一种去中心化、高效的数据存储和检索方法变得尤为重要。

本文设计了一种基于B树和B+树与Bloom过滤器的云存储高效查找结构，为解决这一挑战提供了新的思路。该结构的核心在于通过优化数据索引和检索机制，实现云存储系统中数据的快速定位与访问。主要包括以下几个关键组成部分：

**云存储服务器**：作为数据存储的核心，负责数据的物理存储和维护。

**B树索引模块**：B树是一种平衡的多路搜索树，适用于读写操作相对均衡的场景。它可以保持数据的有序性，使得数据的查找、插入和删除操作都非常高效。

**B+树索引模块**：作为B树的变种，B+树将所有的数据都存储在叶子节点，并且叶子节点之间形成了一个链表。这种结构特别适合于范围查询和顺序访问，因为它的叶子节点中包含了所有的关键字和指向相应数据块的指针，且叶子节点按照键值的大小顺序链接在一起，从而便于进行顺序访问和范围查询。

**Bloom过滤器模块**：Bloom过滤器是一种空间效率很高的概率型数据结构，用于判断一个元素是否在一个集合中。它可以用来检测数据是否已经存在于云存储系统中，从而避免不必要的存储操作。Bloom过滤器由一个位数组和一组哈希函数组成，虽然存在一定的误识别率，但不会发生漏报，即如果Bloom过滤器判断元素存在，则元素可能存在；如果判断不存在，则一定不存在。

**用户接口**：提供用户与云存储系统交互的界面，用户可以通过这个接口进行数据的上传、下载、查询等操作。

通过这种结构，云存储系统可以在保持高效率的同时，提高数据检索的准确性和速度。B+树的引入尤其适合于大量数据的读写操作，而Bloom过滤器则为系统提供了快速判断数据存在性的能力，两者结合，使得云存储系统在处理大规模数据时更加高效和稳定。

**发明内容**

**// 待修改**

**本发明的意义及优势：**

// 待修改

**附图说明**

**// 待修改**

图1为本发明系统概述图示。

图2为本发明上传数据机制的工作流程图示。

图3 为本发明查询数据机制的工作流程图示。

图4为本发明修改数据机制的工作流程图示。

图5为本发明构造的区块链结构图示。

图6为在不同份额数及门限值下，陷门管理中心为修改管理者生成陷门份额的时间消耗。

图7为在不同份额数及门限值下，医院服务器验证陷门份额的时间消耗。

图8为在不同份额数及门限值下，医院服务器进行陷门值恢复的时间消耗。

**具体实施方式：**

**// 待修改**

本发明提出的基于区块链和属性密码的可编辑数字医疗共享系统，利用区块链以及IPFS的去中心化以及不可篡改特性为医疗数据存储提供安全保障；利用CP-ABE加密实现患者隐私保护以及一对多的高效数据共享；最后考虑到实际场景下的数据修改需求，引入变色龙哈希函数以及可验证的秘密共享方案，实现去中心化的链上错误数据可更正。下面将通过具体例子，进一步介绍该发明系统。

陷门管理模块中，设计一种可靠的、安全的错误数据修正机制，是本发明基于区块链的数据存储系统的一个重点。其中，首先需要设置一个陷门管理中心负责管理陷门私钥，为有修改需求的数据生成修改陷门。这个陷门管理中心是权威的、可信的。而决定是否能修改的修改管理者则可以是不诚实的，本系统对恶意修改管理者的最高容错能力为t-1。本发明引入变色龙哈希函数以及可验证的秘密共享技术，确保修改机制实现。其中代表总份额数即份额持有者（本发明中为修改管理者）数量，为恢复秘密的门限值（即为恢复秘密所需的最少有效份额数量），具体细节如下：

**变色龙哈希**函数**：**变色龙哈希函数对于陷门持有者来说是一个可逆的哈希函数，但对于其他未知陷门者仍是单向的、不可逆的；一个变色龙哈希函数主要包含以下五个算法：

（1）初始化算法：输入一个安全参数，输出系统公开参数。其中均为大素数且满足，为阶乘法群的生成元。

（2）密钥生成算法：生成哈希公私钥对，其中，哈希私钥由陷门管理中心秘密保管，而哈希公钥则作为系统参数进行公开；

（3）哈希生成算法：输入数据、哈希公钥以及一个随机数，数据所对应的哈希值的计算式为：

； （1）

（4）陷门生成算法：输入数据、随机数、新数据以及哈希私钥，算出新数据对应的随机数值，即陷门值，其算式为：

； （2）

（5）哈希验证算法：陷门值正确性验证算式为：

 ， （3）

本系统中，将传统区块链中默克尔树（Merkle tree）最底层节点所使用的单向哈希函数替换为变色龙哈希函数。在系统初始化阶段，陷门管理中心调用初始化算法以及密钥生成算法得到系统参数以及哈希密钥对，并将作为系统公开参数在链上进行公开。针对于每个医疗元数据，输入IPFS哈希值、哈希随机数以及哈希公钥，调用哈希生成算法，得到医疗元数据所对应的文件哈希值；再将各文件哈希值生成默克尔树保存在区块中（如图3所示）。而医疗服务器（链上节点）可通过患者身份标识、科室以及上传时间三个字段在本地记录表上检索到需要的医疗数据，并通过哈希验证算法验证记录表中所对应的IPFS哈希值是否有效以及检索该IPFS哈希所对应的文件哈希值是否保存在链上，若均有效则可通过该IPFS哈希值向IPFS下载医疗数据。

**可验证的秘密共享方案：**基于单点故障以及避免修改权限过大的考虑，本系统利用可验证的秘密共享技术设计了一种分布式管理陷门值的方案，即陷门生成算法，具体包含如下算法：

（1）份额生成算法：输入一个秘密值S并选择选择个随机数，然后生成最高次项为次的多项式：

， （4）

最后计算份份额，并分发给个修改管理者，同时向链上公布承诺：

； （5）

（2）份额验证算法：份额请求者通过如下等式来验证份额正确性：

， （6）

（3）份额恢复算法：份额请求者在收集到份正确份额后，通过拉格朗日插值公式来恢复秘密值：

； （7）

陷门管理中心收到链上发起的修改请求后，调用陷门生成算法，输入原IPFS哈希值，原哈希随机数，新IPFS哈希值以及哈希私钥生成陷门值；然后将作为秘密值，调用份额生成算法得到陷门份额并通过链下安全通道分发给修改管理者，同时在链上公布承诺。收到陷门份额后，修改管理者查看修改请求中的修改说明，决定是否同意本次修改，若同意则将自己手中陷门份额公布于区块链上。请求修改者可调用份额验证算法来验证份额正确性，等收集到份正确份额后就可调用份额恢复算法来获得陷门值；最后请求修改者修改IPFS哈希值以及哈希随机值字段。

本发明对主要过程进行实验测试，实验环境为VMware Workstation 16 的 Ubuntu 22.04.2LTS 虚 拟 机 上 进 行， Intel® Core™ i7-9750H CPU @2.60GHz，4GB RAM。其测试结果如下：陷门管理中心为修改请求生成陷门花费的时间为0.0169毫秒；服务器做一次哈希验证的时间为0.056毫秒。实验中，将份额总数与门限值之间的关系设定为,范围设定为。在模拟实验中测定修改机制中的几个主要环节进行测试： 陷门管理中心为修改管理者生成陷门份额（份额生成算法，如图6所示）、服务器对公布的陷门份额进行验证（份额验证算法，如图7所示）、服务器收集到份陷门份额后进行陷门值恢复（份额恢复算法，如图8所示）。 实验数据显示，算法耗费时间为毫秒级别，且与份额与门限值大致成正相关。

**参考文献：**

**// 待修改**

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[Z]. 2008.
2. J. Benet. “IPFS—Content addressed versioned P2P file system (DRAFT3),” arXiv:1407.3561, 2014.
3. C. Zhang, Z. Ni, Y. Xu, et al. “A trustworthy industrial data manage ment scheme based on redactable blockchain,” Journal of Parallel and Distributed Computing, 2021, 152: 167-176
4. Chandramouli, A. Choudhury, A. Patra. “A Survey on Perfectly Secure Verifiable Secret-Sharing,” ACM Computing Surveys (CSUR), 2022,54(11s):1-36.
5. L. Zhang, H. Kan, and H. Honglan. “Patient-centered cross-enterprise document sharing and dynamic consent framework using consortium blockchain and ciphertext-policy attribute-based encryption,” Proceedings of the 19th ACM International Conference on Computing Frontiers, 2022.

**说明书附图**

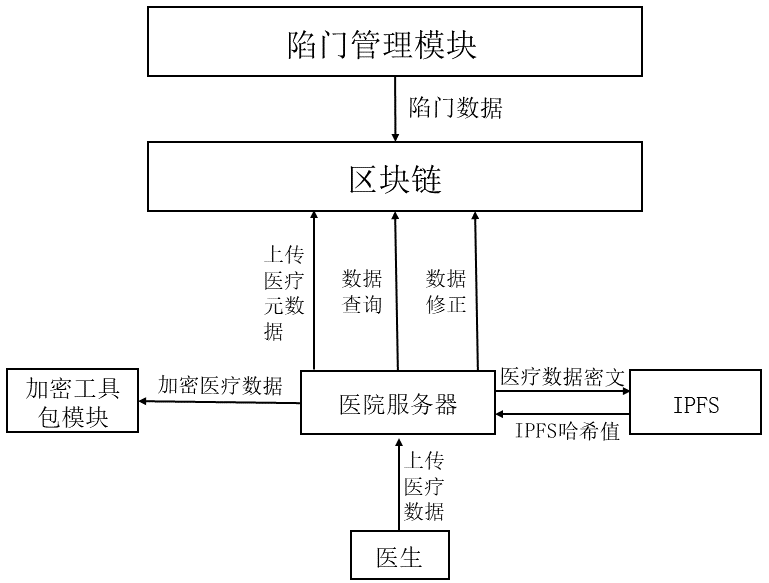


图1

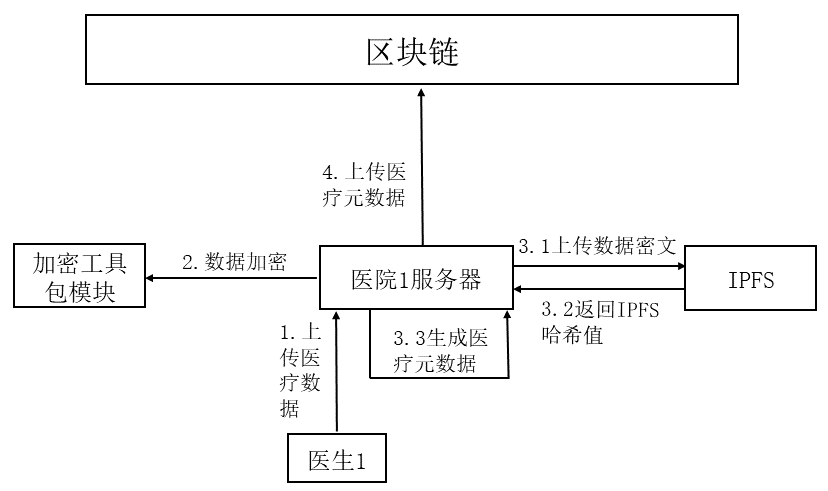
****

图2

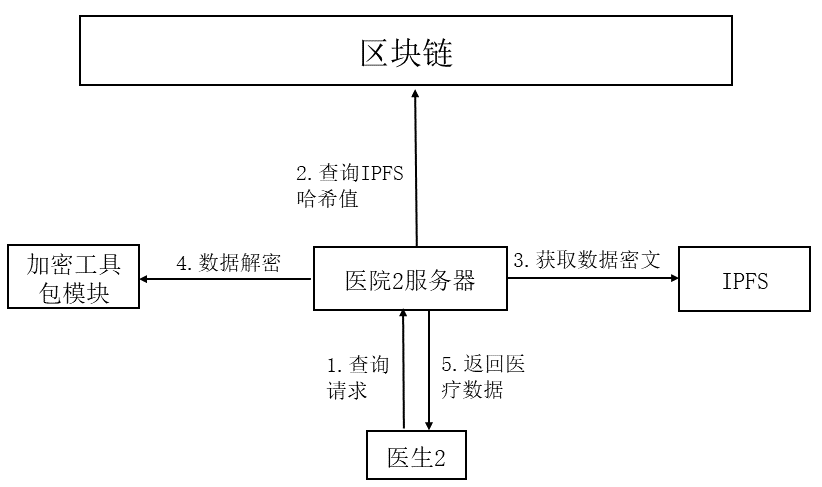


图3

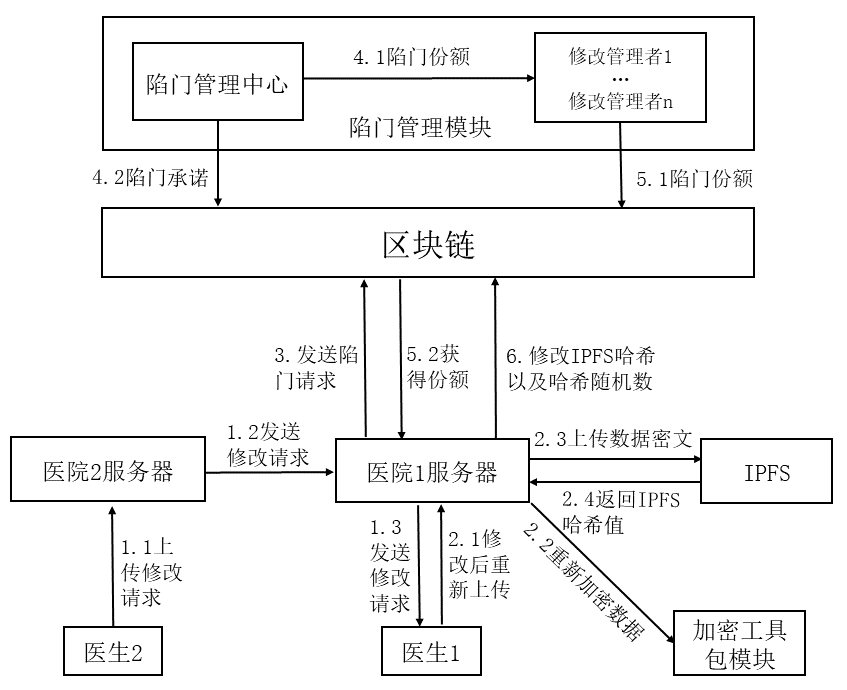


图4

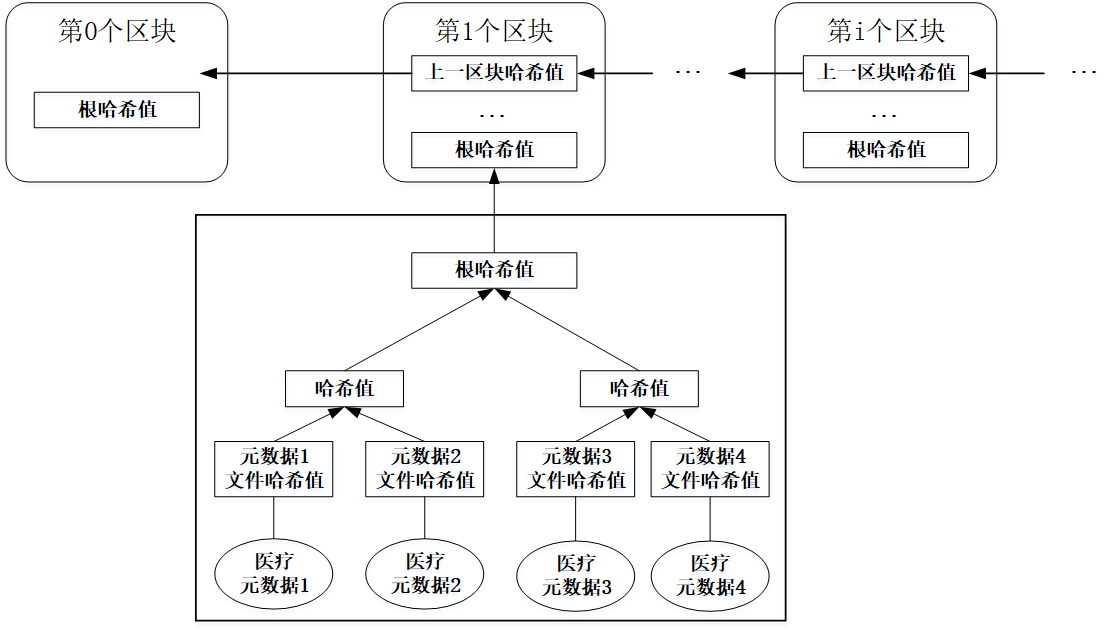


图5

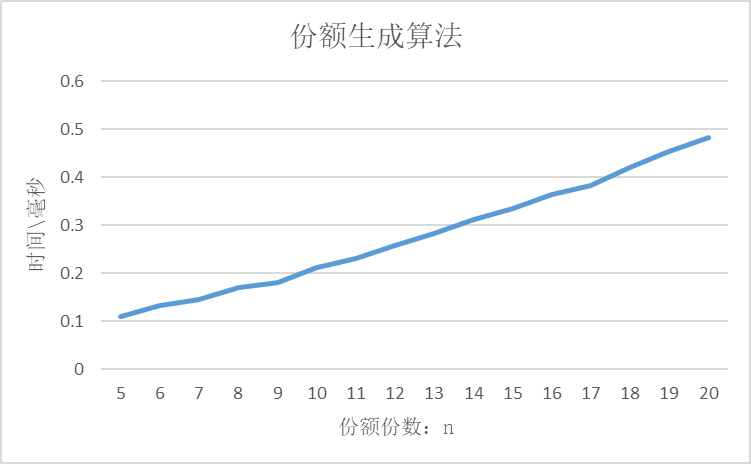


图6

图7

图8