数据结构是一切系统的基础。对于搜索引擎来讲更是非常重要，因为构建搜索引擎所能用到的算法大部分已经成型，非常成熟，而其数据结构可根据具体的需求、具体的硬件设备而出现很多不同的选择。每种数据结构都有其特有的性能指标，最好的却不一定是最适合的，因此在设计搜索引擎时要根据具体需要，选择一种适合的数据结构。

　　一、存储结构

　　目前主流的搜索引擎能够检索的文档数址，基本都在10亿以上，如此巨大的文档如何存储。并且如何在１秒内返回搜索结果。确实是一项挑战，那么一个好的存储结构便是这一切能够得以实到的基础。

　　１、顺序存储方法

顺序存储方法是将数据在物理位置上进行连续的存储。顺序存储的数据都是相邻且连续存放的，因此能够换来很高的扫描速度。但其随机存取效率很低，所以对一些相对固定的不易发生改变的数据应当采取顺序存储方法。

　　２、链接存储方法

链接存储方法不要求数据在物理位置上连续存放。各个数据节点之间用指针进行连接。

　　链接存储相对于顺序存储来讲，不需要事先开辟一整块存储空间，所以，提高了存储空间的利用率，但是扫描数据时，效率比顺序存储要低。

　　３、索引存储方法

索引表由若干索引项组成。若每个节点在索引表中都有一个索引项，则该索引表被称为稠密索引（Dense Index　）。若一组节点在索引表中只对应一个索引项，则该索引表称为稀疏索引（Spare　Index　）。索引项的一般形式是关键字、地址。在实际应用中。有时需要按某些关键字的值查找记录，为此可按关键字建立索引，这种索引叫倒排索引，带有倒排索引的文件叫倒排索引文件，又称为倒排文件。倒排文件可以实现快速检索，这种索引存储方法是目前搜索引擎最常用的存储方法，尤其是倒排索引更是搜索引擎的核心内容。

　　４、散列存储方法

该方法的基本思想是根据节点的关键字直接计算出该节点的存储地址。

　　散列存储类似于散列表，即根据记录中的关键字特点设计一种散列函数（也叫Hash函数）和处理冲突的方法来确定记录的存储位置。将记录散列在存储介质上。这样的文件被称做散列文件。散列文件的随机存取效率很高，对于关键字的值。可以直接由散列函数及冲突处理方法求得在存储介质上的存储位置，从而方便地对它存取。但散列文件不适宜顺序存取和成批处理。

　　４种基本存储方法既可单独使用，也可组合使用。同一逻辑结构采用不同的存储方法，可以得到不同的存取结构。选择何种存储结构来表示相应的逻辑结构，视具体要求而定，主要考虑运算方便及算法的时间要求。

　　二、信息库信息

　　库用来存放所获取的网页。在Google的信息库中需要包含每个网页的HTML文档。其中每个页面都通过Zlib算法进行压缩。在选择采用具体压缩算法的时候，要综合考虑速度和压缩率的关系，一般是它们的折中选择。Zlib的压缩率一般为３：１。

　　Zlib算法是以霍夫曼树编码法和LZ77编码法为编码基础，采用了统计模型和字典模型。实际上是霍夫曼算法与LZ77算法的改进算法。它根据两算法的优缺点能够互补的特点，将两算法的优点有机的结合起来，使压缩效果更好。

　　Zlib算法继承了字典压缩算法的思想。将此思想与滑动窗口（在内存中开辟的一个固定长度的缓冲区）相结合，把滑动窗口看成是字符匹配的字典，找出超前查看缓冲区（与滑动窗口的缓冲区相邻）中能与滑动窗口的字符串相匹配的最长的字符串。并将此串按照LZ77编码法进行编码。当两个缓冲区中没有相匹配的字符时，便采用自适应的霍夫曼编码法进行编码。在编码的过程中将已编好的压缩码实时地写人压缩文件中。解码过程是编码的逆过程，同样用到霍夫曼算法和LZ77算法的编码算法，工作过程与编码过程相似。

　　为了便于从信息库中进行信息的查找，需要对这些信息建立索引。对于搜索到的巨大的信息量来说，必须建立一种合适的、紧凑的数据结构来存放索引。

　　三、文本索引

　　文本索引需要按照一定的次序来保存每个文档的信息，以便于信息的查找。在Google中利用了固定长度的ISAM（索引序列访问模式）进行索引，该索引按照docID排序。在每个索引条目中包含当前文本的状态、一个指向信息库的指针、一个文本的检查值和一些统计信息。如果文本已经被取回，则该条目还包含指向一个可变长度文件的指针，该文件包含文本的URL和标题；否则的话办证，该指针只指向一个URLlist（只包含URL）。这种数据结构可以保证数据的紧凑性和在搜索过程中能够通过一次寻道就取回记录。

　　另外，还有一个用来将URL转换成docID的对照文件，该文件包含了URL校验值和它相应的docID，该文件按照URL的校验值排序。为了能够方便地根据URL找到对应的docID，首先计算该URL的校验值，然后根据二分查找算法在校验值文件中查找对应的docID。一些URL可以批虽地被转换成对应的docID。这种批量转换算法是必要的，因为如果对每个URL都进行一次查找，那么对于一个包含３亿个URL的URL集合来说，系统将要花费大约一个月的时间来进行这种转换。