1.假设我们是爬虫系统 抓取到了两篇文章

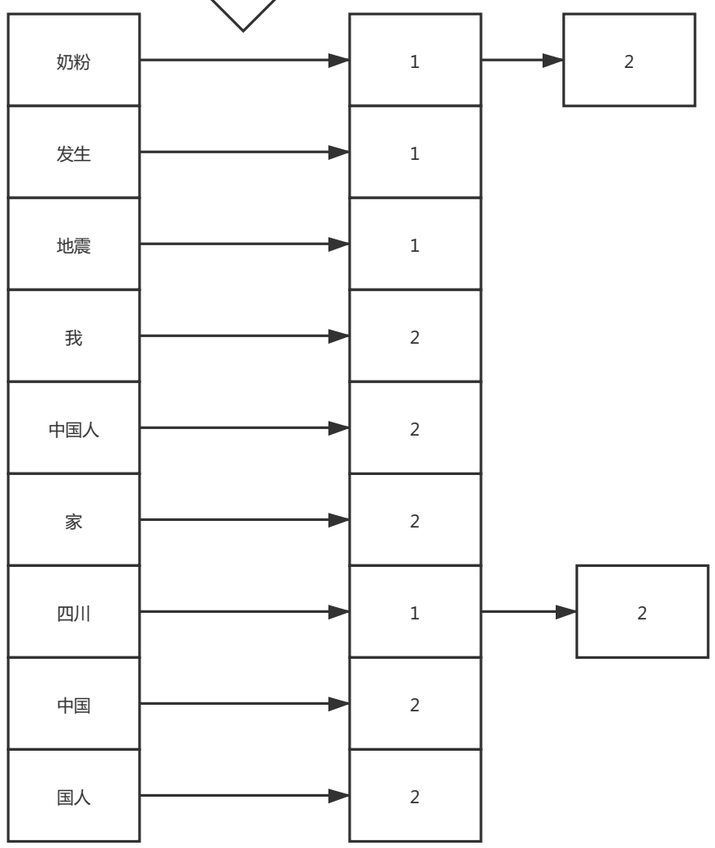
docId：1，内容：四川省发生了地震
docId：2，内容：我是中国人，我家在四川省

分词：四川省/发生/了/地震
分词还会给你过滤：停用词=》四川省/四川/发生/地震

我/中国人/中国/国人/我/家/四川/四川省

倒排索引：
和我们的Hash很像：
key->value
其实在倒排里面是倒过来的
value->key

查字典：
1.我把字典从头翻到尾。
2.目录：直接定位到那一页去



用户检索：美丽的四川省
第一步：分词 美丽/四川省
第二步：把分词的结果去索引里面查找
我们发现四川省有：docId为1 和2的
美丽 并没有：
时间复杂度：O(1) + O(1)
高效也是有代价的 会先把倒排索引结构加载到内存中。数据量很大的时候会采取热数据加载。并不会全部加载，当然你内内存够就会往最大给你加载。
假设我要检索的是：美丽的四川
分词：美丽/四川
漂亮的四川->
上节课讲的一个分词规则：smart和最小颗粒（IK分词器）
武汉市长江大桥：

Smart：武汉市 长江大桥

最小颗粒：武汉 武汉市 长江 市长 长江 长江大桥 江大桥 大桥
以上这两个方式怎么用？是不是就是我们都使用最小颗粒去分词呢？那这个smart有什么意义？
smart找不到 我在最小颗粒 那完了 你的倒排索引 得重建。
搜索的时候使用 smart
建索引的用最小颗粒。
为什么都不全采用最小颗粒呢？
为什么不映射：会用在同义词。相同->相似
假设有几千万的文档包含四川。
链表肯定会很长 会产生哪些问题？
1.存储：压缩问题。1112222335444=>3162231534,Lucene的源码解析
2.取数据：链表怎么取数据呢？从头遍历 比如有1000个数据 要你拿 第500，
遍历500次。怎么办？跳表
3.分页问题：
for()
会加载到内存里面来的 让你找第30万。内存会爆炸 这时候通常的做法就是限制100页 或者 根据业务限制的更小。
所有的搜索引擎没有全部数据导出功能
排序：TF-IDF

英文分分词：
在es里面是有的
english分词器

my name's zhaoyun ,I like eat apples，I am is the xx , running
第一步：空格加标点符号
2.转小写：
3.去停用词
4.会做一个标准化处理
my name zhaoyun I like eat apple run
搜apples 会变成搜 apple
搜apple的时候也是搜 apple