Inverted File Index

以单词或短语作为关键词，将他们出现在文档中的位置记录在索引中

倒排索引：

No Term Times; (Doc ID; Places)

↑ Term dictionary ↑ Posting list

Index generator

```
while (read a document D) {
    while (read a Term T in D) {
        if (Find (Dictionary, T) == false)
            Insert (Dictionary, T);
        Get T's Posting List;
        Insert a node to T's posting list;
    }
}
Write the inverted index to disk;
```

read a term

word stemming 词干分析

process, processing, processed, processes => process.

stop words 停用词

"a" "the" "it". 但会有问题..?

access a term

search trees (B- tree, B+ tree, Tries...)

Hashing

⇒ 如果没有足够的内存:

```
if (out of memory) {
    write blockIndex[cnt] to disk;
    cnt++
    Free Memoryblock;
}
```

最后 merge (Invertedindex, BlockIndex[i])
                                  ↑
                                  0 - cnt

distributed indexing 分布式索引.

Term - partitioned index : a-c, d-f, ...

Document - partitioned index : 文档/机器分布

Dynamic indexing

当有文档 come in, 有 term 要添加或插入, 那么这一块 posting list 是改变的 (dynamic). 另外一部分做上标记, 如果不常用的话 → 相对稳定, main index.
为快来减少操作复杂度. Docs deleted.

Main Index 主的 Auxiliary index 辅助的指

比如说：个莎士比亚文学　　个莫言文学。