**索引文件格式**

索引文件的扩展名是 ".index"

索引文件读取通过定义在 IndexIVF\_HNSW.cpp 里面的函数

**void** **IndexIVF\_HNSW::read**(**const** **char** \*path\_index)

函数代码比较简单，从代码可以看出，索引文件格式如下（表格中的行在文件中顺序存放）：

|  |  |
| --- | --- |
| 组成 | 说明 |
| 向量维度 | size\_t 类型，8 字节 |
| KNN 算法使用的聚类数，就是 centriod 数 | size\_t 类型，8 字节 |
| 向量索引 | 每个 centriod，需要一个向量描述当前 centriod 里面各个点的索引值，这里的向量维度就是centriod里面的点数  这里包含了所有的centriod的索引  每个索引当作一条记录  这些记录顺序存放  每条记录的格式都是：  4 字节的长度信息，表示 centriod 里面的点数  和长度信息匹配的每个点的索引值（32bit整数构成的向量，每一维对应一个点） |
| PQ 编码 | 格式和上面类似，不同之处在于这里是保存残差，所以给PQ量化器残差分配的编码只有8bit，也就是一个字节 |
| 归一化 PQ 编码索引 | 格式和PQ编码器部分完全一样 |
| 归一化粗糙 PQ 编码时，  每个 centriod 到整个空间重心的归一化欧几里德距离 | 前面的向量都是针对每个点有一维数据，这个是针对centroid的，每个centroid到整个空间重心的归一化欧几里德距离 |

读取索引文件的代码位于：

tests/test\_ivfhnsw\_deep1b.cpp: index->read(opt.path\_index);

tests/test\_ivfhnsw\_grouping\_deep1b.cpp: index->read(opt.path\_index);

tests/test\_ivfhnsw\_grouping\_sift1b.cpp: index->read(opt.path\_index);

tests/test\_ivfhnsw\_sift1b.cpp: index->read(opt.path\_index);

TODO:

根据这里的描述，编写索引文件验证工具，如果发现不符合上面的描述，报告错误，以便修正本文档，或者修正验证工具