package com.twitter.search.core.earlybird.index;

/\*\*

\* A doc ID mapper that assigns doc IDs sequentially in decreasing order, starting with the given

\* max ID. Used by Expertsearch, which doesn't index tweets.

\*/

public class SequentialDocIDMapper implements DocIDToTweetIDMapper {

private final int maxSegmentSize;

private int lastAssignedDocId;

public SequentialDocIDMapper(int maxSegmentSize) {

this.maxSegmentSize = maxSegmentSize;

lastAssignedDocId = maxSegmentSize;

}

@Override

public long getTweetID(int docID) {

// Should be used only at segment optimization time and in tests.

if ((docID < lastAssignedDocId) || (docID >= maxSegmentSize)) {

return ID\_NOT\_FOUND;

}

return docID;

}

@Override

public int getDocID(long tweetID) {

// Should be used only at segment optimization time and in tests.

if ((tweetID < lastAssignedDocId) || (tweetID >= maxSegmentSize)) {

return ID\_NOT\_FOUND;

}

return (int) tweetID;

}

@Override

public int getNumDocs() {

return maxSegmentSize - lastAssignedDocId;

}

@Override

public int getNextDocID(int docID) {

int nextDocID = docID + 1;

// nextDocID is larger than any doc ID that can be assigned by this mapper.

if (nextDocID >= maxSegmentSize) {

return ID\_NOT\_FOUND;

}

// nextDocID is smaller than any doc ID assigned by this mapper so far.

if (nextDocID < lastAssignedDocId) {

return lastAssignedDocId;

}

// nextDocID is in the range of doc IDs assigned by this mapper.

return nextDocID;

}

@Override

public int getPreviousDocID(int docID) {

int previousDocID = docID - 1;

// previousDocID is larger than any doc ID that can be assigned by this mapper.

if (previousDocID >= maxSegmentSize) {

return maxSegmentSize - 1;

}

// previousDocID is smaller than any doc ID assigned by this mapper so far.

if (previousDocID < lastAssignedDocId) {

return ID\_NOT\_FOUND;

}

// previousDocID is in the range of doc IDs assigned by this mapper.

return previousDocID;

}

@Override

public int addMapping(final long tweetID) {

return --lastAssignedDocId;

}

@Override

public DocIDToTweetIDMapper optimize() {

// Segments that use this DocIDToTweetIDMapper should never be optimized.

throw new UnsupportedOperationException("SequentialDocIDMapper cannot be optimized.");

}

}