package com.twitter.search.common.search;

import java.util.Arrays;

import org.apache.lucene.search.DocIdSetIterator;

/\*\*

\* DocIdSetIterator implementation from a sorted list of non-negative integers. If the given list of

\* doc IDs is not sorted or contains negative doc IDs, the results are undefined.

\*/

public class IntArrayDocIdSetIterator extends DocIdSetIterator {

private final int[] docIds;

private int docId;

private int cursor;

public IntArrayDocIdSetIterator(int[] ids) {

docIds = ids;

reset();

}

/\*\* Used for testing. \*/

public void reset() {

docId = -1;

cursor = -1;

}

@Override

public int docID() {

return docId;

}

@Override

public int nextDoc() {

return advance(docId);

}

@Override

public int advance(int target) {

if (docId == NO\_MORE\_DOCS) {

return docId;

}

if (target < docId) {

return docId;

}

if (cursor == docIds.length - 1) {

docId = NO\_MORE\_DOCS;

return docId;

}

if (target == docId) {

docId = docIds[++cursor];

return docId;

}

int toIndex = Math.min(cursor + (target - docId) + 1, docIds.length);

int targetIndex = Arrays.binarySearch(docIds, cursor + 1, toIndex, target);

if (targetIndex < 0) {

targetIndex = -targetIndex - 1;

}

if (targetIndex == docIds.length) {

docId = NO\_MORE\_DOCS;

} else {

cursor = targetIndex;

docId = docIds[cursor];

}

return docId;

}

@Override

public long cost() {

return docIds == null ? 0 : docIds.length;

}

}