/\* ----------------------------------------------------------------------------

\* This file was automatically generated by SWIG (http://www.swig.org).

\* Version 4.0.2

\*

\* Do not make changes to this file unless you know what you are doing--modify

\* the SWIG interface file instead.

\* ----------------------------------------------------------------------------- \*/

package com.twitter.ann.faiss;

public class IndexBinaryIVF extends IndexBinary {

private transient long swigCPtr;

protected IndexBinaryIVF(long cPtr, boolean cMemoryOwn) {

super(swigfaissJNI.IndexBinaryIVF\_SWIGUpcast(cPtr), cMemoryOwn);

swigCPtr = cPtr;

}

protected static long getCPtr(IndexBinaryIVF obj) {

return (obj == null) ? 0 : obj.swigCPtr;

}

@SuppressWarnings("deprecation")

protected void finalize() {

delete();

}

public synchronized void delete() {

if (swigCPtr != 0) {

if (swigCMemOwn) {

swigCMemOwn = false;

swigfaissJNI.delete\_IndexBinaryIVF(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setInvlists(InvertedLists value) {

swigfaissJNI.IndexBinaryIVF\_invlists\_set(swigCPtr, this, InvertedLists.getCPtr(value), value);

}

public InvertedLists getInvlists() {

long cPtr = swigfaissJNI.IndexBinaryIVF\_invlists\_get(swigCPtr, this);

return (cPtr == 0) ? null : new InvertedLists(cPtr, false);

}

public void setOwn\_invlists(boolean value) {

swigfaissJNI.IndexBinaryIVF\_own\_invlists\_set(swigCPtr, this, value);

}

public boolean getOwn\_invlists() {

return swigfaissJNI.IndexBinaryIVF\_own\_invlists\_get(swigCPtr, this);

}

public void setNprobe(long value) {

swigfaissJNI.IndexBinaryIVF\_nprobe\_set(swigCPtr, this, value);

}

public long getNprobe() {

return swigfaissJNI.IndexBinaryIVF\_nprobe\_get(swigCPtr, this);

}

public void setMax\_codes(long value) {

swigfaissJNI.IndexBinaryIVF\_max\_codes\_set(swigCPtr, this, value);

}

public long getMax\_codes() {

return swigfaissJNI.IndexBinaryIVF\_max\_codes\_get(swigCPtr, this);

}

public void setUse\_heap(boolean value) {

swigfaissJNI.IndexBinaryIVF\_use\_heap\_set(swigCPtr, this, value);

}

public boolean getUse\_heap() {

return swigfaissJNI.IndexBinaryIVF\_use\_heap\_get(swigCPtr, this);

}

public void setDirect\_map(SWIGTYPE\_p\_DirectMap value) {

swigfaissJNI.IndexBinaryIVF\_direct\_map\_set(swigCPtr, this, SWIGTYPE\_p\_DirectMap.getCPtr(value));

}

public SWIGTYPE\_p\_DirectMap getDirect\_map() {

return new SWIGTYPE\_p\_DirectMap(swigfaissJNI.IndexBinaryIVF\_direct\_map\_get(swigCPtr, this), true);

}

public void setQuantizer(IndexBinary value) {

swigfaissJNI.IndexBinaryIVF\_quantizer\_set(swigCPtr, this, IndexBinary.getCPtr(value), value);

}

public IndexBinary getQuantizer() {

long cPtr = swigfaissJNI.IndexBinaryIVF\_quantizer\_get(swigCPtr, this);

return (cPtr == 0) ? null : new IndexBinary(cPtr, false);

}

public void setNlist(long value) {

swigfaissJNI.IndexBinaryIVF\_nlist\_set(swigCPtr, this, value);

}

public long getNlist() {

return swigfaissJNI.IndexBinaryIVF\_nlist\_get(swigCPtr, this);

}

public void setOwn\_fields(boolean value) {

swigfaissJNI.IndexBinaryIVF\_own\_fields\_set(swigCPtr, this, value);

}

public boolean getOwn\_fields() {

return swigfaissJNI.IndexBinaryIVF\_own\_fields\_get(swigCPtr, this);

}

public void setCp(ClusteringParameters value) {

swigfaissJNI.IndexBinaryIVF\_cp\_set(swigCPtr, this, ClusteringParameters.getCPtr(value), value);

}

public ClusteringParameters getCp() {

long cPtr = swigfaissJNI.IndexBinaryIVF\_cp\_get(swigCPtr, this);

return (cPtr == 0) ? null : new ClusteringParameters(cPtr, false);

}

public void setClustering\_index(Index value) {

swigfaissJNI.IndexBinaryIVF\_clustering\_index\_set(swigCPtr, this, Index.getCPtr(value), value);

}

public Index getClustering\_index() {

long cPtr = swigfaissJNI.IndexBinaryIVF\_clustering\_index\_get(swigCPtr, this);

return (cPtr == 0) ? null : new Index(cPtr, false);

}

public IndexBinaryIVF(IndexBinary quantizer, long d, long nlist) {

this(swigfaissJNI.new\_IndexBinaryIVF\_\_SWIG\_0(IndexBinary.getCPtr(quantizer), quantizer, d, nlist), true);

}

public IndexBinaryIVF() {

this(swigfaissJNI.new\_IndexBinaryIVF\_\_SWIG\_1(), true);

}

public void reset() {

swigfaissJNI.IndexBinaryIVF\_reset(swigCPtr, this);

}

public void train(long n, SWIGTYPE\_p\_unsigned\_char x) {

swigfaissJNI.IndexBinaryIVF\_train(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x));

}

public void add(long n, SWIGTYPE\_p\_unsigned\_char x) {

swigfaissJNI.IndexBinaryIVF\_add(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x));

}

public void add\_with\_ids(long n, SWIGTYPE\_p\_unsigned\_char x, LongVector xids) {

swigfaissJNI.IndexBinaryIVF\_add\_with\_ids(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), SWIGTYPE\_p\_long\_long.getCPtr(xids.data()), xids);

}

public void add\_core(long n, SWIGTYPE\_p\_unsigned\_char x, LongVector xids, LongVector precomputed\_idx) {

swigfaissJNI.IndexBinaryIVF\_add\_core(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), SWIGTYPE\_p\_long\_long.getCPtr(xids.data()), xids, SWIGTYPE\_p\_long\_long.getCPtr(precomputed\_idx.data()), precomputed\_idx);

}

public void search\_preassigned(long n, SWIGTYPE\_p\_unsigned\_char x, long k, LongVector assign, SWIGTYPE\_p\_int centroid\_dis, SWIGTYPE\_p\_int distances, LongVector labels, boolean store\_pairs, IVFSearchParameters params) {

swigfaissJNI.IndexBinaryIVF\_search\_preassigned\_\_SWIG\_0(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), k, SWIGTYPE\_p\_long\_long.getCPtr(assign.data()), assign, SWIGTYPE\_p\_int.getCPtr(centroid\_dis), SWIGTYPE\_p\_int.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, store\_pairs, IVFSearchParameters.getCPtr(params), params);

}

public void search\_preassigned(long n, SWIGTYPE\_p\_unsigned\_char x, long k, LongVector assign, SWIGTYPE\_p\_int centroid\_dis, SWIGTYPE\_p\_int distances, LongVector labels, boolean store\_pairs) {

swigfaissJNI.IndexBinaryIVF\_search\_preassigned\_\_SWIG\_1(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), k, SWIGTYPE\_p\_long\_long.getCPtr(assign.data()), assign, SWIGTYPE\_p\_int.getCPtr(centroid\_dis), SWIGTYPE\_p\_int.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, store\_pairs);

}

public SWIGTYPE\_p\_faiss\_\_BinaryInvertedListScanner get\_InvertedListScanner(boolean store\_pairs) {

long cPtr = swigfaissJNI.IndexBinaryIVF\_get\_InvertedListScanner\_\_SWIG\_0(swigCPtr, this, store\_pairs);

return (cPtr == 0) ? null : new SWIGTYPE\_p\_faiss\_\_BinaryInvertedListScanner(cPtr, false);

}

public SWIGTYPE\_p\_faiss\_\_BinaryInvertedListScanner get\_InvertedListScanner() {

long cPtr = swigfaissJNI.IndexBinaryIVF\_get\_InvertedListScanner\_\_SWIG\_1(swigCPtr, this);

return (cPtr == 0) ? null : new SWIGTYPE\_p\_faiss\_\_BinaryInvertedListScanner(cPtr, false);

}

public void search(long n, SWIGTYPE\_p\_unsigned\_char x, long k, SWIGTYPE\_p\_int distances, LongVector labels) {

swigfaissJNI.IndexBinaryIVF\_search(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), k, SWIGTYPE\_p\_int.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels);

}

public void range\_search(long n, SWIGTYPE\_p\_unsigned\_char x, int radius, RangeSearchResult result) {

swigfaissJNI.IndexBinaryIVF\_range\_search(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), radius, RangeSearchResult.getCPtr(result), result);

}

public void range\_search\_preassigned(long n, SWIGTYPE\_p\_unsigned\_char x, int radius, LongVector assign, SWIGTYPE\_p\_int centroid\_dis, RangeSearchResult result) {

swigfaissJNI.IndexBinaryIVF\_range\_search\_preassigned(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), radius, SWIGTYPE\_p\_long\_long.getCPtr(assign.data()), assign, SWIGTYPE\_p\_int.getCPtr(centroid\_dis), RangeSearchResult.getCPtr(result), result);

}

public void reconstruct(long key, SWIGTYPE\_p\_unsigned\_char recons) {

swigfaissJNI.IndexBinaryIVF\_reconstruct(swigCPtr, this, key, SWIGTYPE\_p\_unsigned\_char.getCPtr(recons));

}

public void reconstruct\_n(long i0, long ni, SWIGTYPE\_p\_unsigned\_char recons) {

swigfaissJNI.IndexBinaryIVF\_reconstruct\_n(swigCPtr, this, i0, ni, SWIGTYPE\_p\_unsigned\_char.getCPtr(recons));

}

public void search\_and\_reconstruct(long n, SWIGTYPE\_p\_unsigned\_char x, long k, SWIGTYPE\_p\_int distances, LongVector labels, SWIGTYPE\_p\_unsigned\_char recons) {

swigfaissJNI.IndexBinaryIVF\_search\_and\_reconstruct(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(x), k, SWIGTYPE\_p\_int.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, SWIGTYPE\_p\_unsigned\_char.getCPtr(recons));

}

public void reconstruct\_from\_offset(long list\_no, long offset, SWIGTYPE\_p\_unsigned\_char recons) {

swigfaissJNI.IndexBinaryIVF\_reconstruct\_from\_offset(swigCPtr, this, list\_no, offset, SWIGTYPE\_p\_unsigned\_char.getCPtr(recons));

}

public long remove\_ids(IDSelector sel) {

return swigfaissJNI.IndexBinaryIVF\_remove\_ids(swigCPtr, this, IDSelector.getCPtr(sel), sel);

}

public void merge\_from(IndexBinaryIVF other, long add\_id) {

swigfaissJNI.IndexBinaryIVF\_merge\_from(swigCPtr, this, IndexBinaryIVF.getCPtr(other), other, add\_id);

}

public long get\_list\_size(long list\_no) {

return swigfaissJNI.IndexBinaryIVF\_get\_list\_size(swigCPtr, this, list\_no);

}

public void make\_direct\_map(boolean new\_maintain\_direct\_map) {

swigfaissJNI.IndexBinaryIVF\_make\_direct\_map\_\_SWIG\_0(swigCPtr, this, new\_maintain\_direct\_map);

}

public void make\_direct\_map() {

swigfaissJNI.IndexBinaryIVF\_make\_direct\_map\_\_SWIG\_1(swigCPtr, this);

}

public void set\_direct\_map\_type(SWIGTYPE\_p\_DirectMap\_\_Type type) {

swigfaissJNI.IndexBinaryIVF\_set\_direct\_map\_type(swigCPtr, this, SWIGTYPE\_p\_DirectMap\_\_Type.getCPtr(type));

}

public void replace\_invlists(InvertedLists il, boolean own) {

swigfaissJNI.IndexBinaryIVF\_replace\_invlists\_\_SWIG\_0(swigCPtr, this, InvertedLists.getCPtr(il), il, own);

}

public void replace\_invlists(InvertedLists il) {

swigfaissJNI.IndexBinaryIVF\_replace\_invlists\_\_SWIG\_1(swigCPtr, this, InvertedLists.getCPtr(il), il);

}

}