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

\* 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 IndexIVF extends Index {

private transient long swigCPtr;

protected IndexIVF(long cPtr, boolean cMemoryOwn) {

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

swigCPtr = cPtr;

}

protected static long getCPtr(IndexIVF 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\_IndexIVF(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setInvlists(InvertedLists value) {

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

}

public InvertedLists getInvlists() {

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

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

}

public void setOwn\_invlists(boolean value) {

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

}

public boolean getOwn\_invlists() {

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

}

public void setCode\_size(long value) {

swigfaissJNI.IndexIVF\_code\_size\_set(swigCPtr, this, value);

}

public long getCode\_size() {

return swigfaissJNI.IndexIVF\_code\_size\_get(swigCPtr, this);

}

public void setNprobe(long value) {

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

}

public long getNprobe() {

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

}

public void setMax\_codes(long value) {

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

}

public long getMax\_codes() {

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

}

public void setParallel\_mode(int value) {

swigfaissJNI.IndexIVF\_parallel\_mode\_set(swigCPtr, this, value);

}

public int getParallel\_mode() {

return swigfaissJNI.IndexIVF\_parallel\_mode\_get(swigCPtr, this);

}

public int getPARALLEL\_MODE\_NO\_HEAP\_INIT() {

return swigfaissJNI.IndexIVF\_PARALLEL\_MODE\_NO\_HEAP\_INIT\_get(swigCPtr, this);

}

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

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

}

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

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

}

public void reset() {

swigfaissJNI.IndexIVF\_reset(swigCPtr, this);

}

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

swigfaissJNI.IndexIVF\_train(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x));

}

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

swigfaissJNI.IndexIVF\_add(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x));

}

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

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

}

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

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

}

public void encode\_vectors(long n, SWIGTYPE\_p\_float x, LongVector list\_nos, SWIGTYPE\_p\_unsigned\_char codes, boolean include\_listno) {

swigfaissJNI.IndexIVF\_encode\_vectors\_\_SWIG\_0(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), SWIGTYPE\_p\_long\_long.getCPtr(list\_nos.data()), list\_nos, SWIGTYPE\_p\_unsigned\_char.getCPtr(codes), include\_listno);

}

public void encode\_vectors(long n, SWIGTYPE\_p\_float x, LongVector list\_nos, SWIGTYPE\_p\_unsigned\_char codes) {

swigfaissJNI.IndexIVF\_encode\_vectors\_\_SWIG\_1(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), SWIGTYPE\_p\_long\_long.getCPtr(list\_nos.data()), list\_nos, SWIGTYPE\_p\_unsigned\_char.getCPtr(codes));

}

public void add\_sa\_codes(long n, SWIGTYPE\_p\_unsigned\_char codes, LongVector xids) {

swigfaissJNI.IndexIVF\_add\_sa\_codes(swigCPtr, this, n, SWIGTYPE\_p\_unsigned\_char.getCPtr(codes), SWIGTYPE\_p\_long\_long.getCPtr(xids.data()), xids);

}

public void train\_residual(long n, SWIGTYPE\_p\_float x) {

swigfaissJNI.IndexIVF\_train\_residual(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x));

}

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

swigfaissJNI.IndexIVF\_search\_preassigned\_\_SWIG\_0(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), k, SWIGTYPE\_p\_long\_long.getCPtr(assign.data()), assign, SWIGTYPE\_p\_float.getCPtr(centroid\_dis), SWIGTYPE\_p\_float.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, store\_pairs, IVFSearchParameters.getCPtr(params), params, IndexIVFStats.getCPtr(stats), stats);

}

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

swigfaissJNI.IndexIVF\_search\_preassigned\_\_SWIG\_1(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), k, SWIGTYPE\_p\_long\_long.getCPtr(assign.data()), assign, SWIGTYPE\_p\_float.getCPtr(centroid\_dis), SWIGTYPE\_p\_float.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, store\_pairs, IVFSearchParameters.getCPtr(params), params);

}

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

swigfaissJNI.IndexIVF\_search\_preassigned\_\_SWIG\_2(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), k, SWIGTYPE\_p\_long\_long.getCPtr(assign.data()), assign, SWIGTYPE\_p\_float.getCPtr(centroid\_dis), SWIGTYPE\_p\_float.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, store\_pairs);

}

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

swigfaissJNI.IndexIVF\_search(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), k, SWIGTYPE\_p\_float.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels);

}

public void range\_search(long n, SWIGTYPE\_p\_float x, float radius, RangeSearchResult result) {

swigfaissJNI.IndexIVF\_range\_search(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), radius, RangeSearchResult.getCPtr(result), result);

}

public void range\_search\_preassigned(long nx, SWIGTYPE\_p\_float x, float radius, LongVector keys, SWIGTYPE\_p\_float coarse\_dis, RangeSearchResult result, boolean store\_pairs, IVFSearchParameters params, IndexIVFStats stats) {

swigfaissJNI.IndexIVF\_range\_search\_preassigned\_\_SWIG\_0(swigCPtr, this, nx, SWIGTYPE\_p\_float.getCPtr(x), radius, SWIGTYPE\_p\_long\_long.getCPtr(keys.data()), keys, SWIGTYPE\_p\_float.getCPtr(coarse\_dis), RangeSearchResult.getCPtr(result), result, store\_pairs, IVFSearchParameters.getCPtr(params), params, IndexIVFStats.getCPtr(stats), stats);

}

public void range\_search\_preassigned(long nx, SWIGTYPE\_p\_float x, float radius, LongVector keys, SWIGTYPE\_p\_float coarse\_dis, RangeSearchResult result, boolean store\_pairs, IVFSearchParameters params) {

swigfaissJNI.IndexIVF\_range\_search\_preassigned\_\_SWIG\_1(swigCPtr, this, nx, SWIGTYPE\_p\_float.getCPtr(x), radius, SWIGTYPE\_p\_long\_long.getCPtr(keys.data()), keys, SWIGTYPE\_p\_float.getCPtr(coarse\_dis), RangeSearchResult.getCPtr(result), result, store\_pairs, IVFSearchParameters.getCPtr(params), params);

}

public void range\_search\_preassigned(long nx, SWIGTYPE\_p\_float x, float radius, LongVector keys, SWIGTYPE\_p\_float coarse\_dis, RangeSearchResult result, boolean store\_pairs) {

swigfaissJNI.IndexIVF\_range\_search\_preassigned\_\_SWIG\_2(swigCPtr, this, nx, SWIGTYPE\_p\_float.getCPtr(x), radius, SWIGTYPE\_p\_long\_long.getCPtr(keys.data()), keys, SWIGTYPE\_p\_float.getCPtr(coarse\_dis), RangeSearchResult.getCPtr(result), result, store\_pairs);

}

public void range\_search\_preassigned(long nx, SWIGTYPE\_p\_float x, float radius, LongVector keys, SWIGTYPE\_p\_float coarse\_dis, RangeSearchResult result) {

swigfaissJNI.IndexIVF\_range\_search\_preassigned\_\_SWIG\_3(swigCPtr, this, nx, SWIGTYPE\_p\_float.getCPtr(x), radius, SWIGTYPE\_p\_long\_long.getCPtr(keys.data()), keys, SWIGTYPE\_p\_float.getCPtr(coarse\_dis), RangeSearchResult.getCPtr(result), result);

}

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

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

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

}

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

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

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

}

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

swigfaissJNI.IndexIVF\_reconstruct(swigCPtr, this, key, SWIGTYPE\_p\_float.getCPtr(recons));

}

public void update\_vectors(int nv, LongVector idx, SWIGTYPE\_p\_float v) {

swigfaissJNI.IndexIVF\_update\_vectors(swigCPtr, this, nv, SWIGTYPE\_p\_long\_long.getCPtr(idx.data()), idx, SWIGTYPE\_p\_float.getCPtr(v));

}

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

swigfaissJNI.IndexIVF\_reconstruct\_n(swigCPtr, this, i0, ni, SWIGTYPE\_p\_float.getCPtr(recons));

}

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

swigfaissJNI.IndexIVF\_search\_and\_reconstruct(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), k, SWIGTYPE\_p\_float.getCPtr(distances), SWIGTYPE\_p\_long\_long.getCPtr(labels.data()), labels, SWIGTYPE\_p\_float.getCPtr(recons));

}

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

swigfaissJNI.IndexIVF\_reconstruct\_from\_offset(swigCPtr, this, list\_no, offset, SWIGTYPE\_p\_float.getCPtr(recons));

}

public long remove\_ids(IDSelector sel) {

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

}

public void check\_compatible\_for\_merge(IndexIVF other) {

swigfaissJNI.IndexIVF\_check\_compatible\_for\_merge(swigCPtr, this, IndexIVF.getCPtr(other), other);

}

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

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

}

public void copy\_subset\_to(IndexIVF other, int subset\_type, long a1, long a2) {

swigfaissJNI.IndexIVF\_copy\_subset\_to(swigCPtr, this, IndexIVF.getCPtr(other), other, subset\_type, a1, a2);

}

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

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

}

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

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

}

public void make\_direct\_map() {

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

}

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

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

}

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

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

}

public void replace\_invlists(InvertedLists il) {

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

}

public long sa\_code\_size() {

return swigfaissJNI.IndexIVF\_sa\_code\_size(swigCPtr, this);

}

public void sa\_encode(long n, SWIGTYPE\_p\_float x, SWIGTYPE\_p\_unsigned\_char bytes) {

swigfaissJNI.IndexIVF\_sa\_encode(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x), SWIGTYPE\_p\_unsigned\_char.getCPtr(bytes));

}

}