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

\* 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 HNSW {

private transient long swigCPtr;

protected transient boolean swigCMemOwn;

protected HNSW(long cPtr, boolean cMemoryOwn) {

swigCMemOwn = cMemoryOwn;

swigCPtr = cPtr;

}

protected static long getCPtr(HNSW 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\_HNSW(swigCPtr);

}

swigCPtr = 0;

}

}

static public class MinimaxHeap {

private transient long swigCPtr;

protected transient boolean swigCMemOwn;

protected MinimaxHeap(long cPtr, boolean cMemoryOwn) {

swigCMemOwn = cMemoryOwn;

swigCPtr = cPtr;

}

protected static long getCPtr(MinimaxHeap 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\_HNSW\_MinimaxHeap(swigCPtr);

}

swigCPtr = 0;

}

}

public void setN(int value) {

swigfaissJNI.HNSW\_MinimaxHeap\_n\_set(swigCPtr, this, value);

}

public int getN() {

return swigfaissJNI.HNSW\_MinimaxHeap\_n\_get(swigCPtr, this);

}

public void setK(int value) {

swigfaissJNI.HNSW\_MinimaxHeap\_k\_set(swigCPtr, this, value);

}

public int getK() {

return swigfaissJNI.HNSW\_MinimaxHeap\_k\_get(swigCPtr, this);

}

public void setNvalid(int value) {

swigfaissJNI.HNSW\_MinimaxHeap\_nvalid\_set(swigCPtr, this, value);

}

public int getNvalid() {

return swigfaissJNI.HNSW\_MinimaxHeap\_nvalid\_get(swigCPtr, this);

}

public void setIds(IntVector value) {

swigfaissJNI.HNSW\_MinimaxHeap\_ids\_set(swigCPtr, this, IntVector.getCPtr(value), value);

}

public IntVector getIds() {

long cPtr = swigfaissJNI.HNSW\_MinimaxHeap\_ids\_get(swigCPtr, this);

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

}

public void setDis(FloatVector value) {

swigfaissJNI.HNSW\_MinimaxHeap\_dis\_set(swigCPtr, this, FloatVector.getCPtr(value), value);

}

public FloatVector getDis() {

long cPtr = swigfaissJNI.HNSW\_MinimaxHeap\_dis\_get(swigCPtr, this);

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

}

public MinimaxHeap(int n) {

this(swigfaissJNI.new\_HNSW\_MinimaxHeap(n), true);

}

public void push(int i, float v) {

swigfaissJNI.HNSW\_MinimaxHeap\_push(swigCPtr, this, i, v);

}

public float max() {

return swigfaissJNI.HNSW\_MinimaxHeap\_max(swigCPtr, this);

}

public int size() {

return swigfaissJNI.HNSW\_MinimaxHeap\_size(swigCPtr, this);

}

public void clear() {

swigfaissJNI.HNSW\_MinimaxHeap\_clear(swigCPtr, this);

}

public int pop\_min(SWIGTYPE\_p\_float vmin\_out) {

return swigfaissJNI.HNSW\_MinimaxHeap\_pop\_min\_\_SWIG\_0(swigCPtr, this, SWIGTYPE\_p\_float.getCPtr(vmin\_out));

}

public int pop\_min() {

return swigfaissJNI.HNSW\_MinimaxHeap\_pop\_min\_\_SWIG\_1(swigCPtr, this);

}

public int count\_below(float thresh) {

return swigfaissJNI.HNSW\_MinimaxHeap\_count\_below(swigCPtr, this, thresh);

}

}

static public class NodeDistCloser {

private transient long swigCPtr;

protected transient boolean swigCMemOwn;

protected NodeDistCloser(long cPtr, boolean cMemoryOwn) {

swigCMemOwn = cMemoryOwn;

swigCPtr = cPtr;

}

protected static long getCPtr(NodeDistCloser 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\_HNSW\_NodeDistCloser(swigCPtr);

}

swigCPtr = 0;

}

}

public void setD(float value) {

swigfaissJNI.HNSW\_NodeDistCloser\_d\_set(swigCPtr, this, value);

}

public float getD() {

return swigfaissJNI.HNSW\_NodeDistCloser\_d\_get(swigCPtr, this);

}

public void setId(int value) {

swigfaissJNI.HNSW\_NodeDistCloser\_id\_set(swigCPtr, this, value);

}

public int getId() {

return swigfaissJNI.HNSW\_NodeDistCloser\_id\_get(swigCPtr, this);

}

public NodeDistCloser(float d, int id) {

this(swigfaissJNI.new\_HNSW\_NodeDistCloser(d, id), true);

}

}

static public class NodeDistFarther {

private transient long swigCPtr;

protected transient boolean swigCMemOwn;

protected NodeDistFarther(long cPtr, boolean cMemoryOwn) {

swigCMemOwn = cMemoryOwn;

swigCPtr = cPtr;

}

protected static long getCPtr(NodeDistFarther 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\_HNSW\_NodeDistFarther(swigCPtr);

}

swigCPtr = 0;

}

}

public void setD(float value) {

swigfaissJNI.HNSW\_NodeDistFarther\_d\_set(swigCPtr, this, value);

}

public float getD() {

return swigfaissJNI.HNSW\_NodeDistFarther\_d\_get(swigCPtr, this);

}

public void setId(int value) {

swigfaissJNI.HNSW\_NodeDistFarther\_id\_set(swigCPtr, this, value);

}

public int getId() {

return swigfaissJNI.HNSW\_NodeDistFarther\_id\_get(swigCPtr, this);

}

public NodeDistFarther(float d, int id) {

this(swigfaissJNI.new\_HNSW\_NodeDistFarther(d, id), true);

}

}

public void setAssign\_probas(DoubleVector value) {

swigfaissJNI.HNSW\_assign\_probas\_set(swigCPtr, this, DoubleVector.getCPtr(value), value);

}

public DoubleVector getAssign\_probas() {

long cPtr = swigfaissJNI.HNSW\_assign\_probas\_get(swigCPtr, this);

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

}

public void setCum\_nneighbor\_per\_level(IntVector value) {

swigfaissJNI.HNSW\_cum\_nneighbor\_per\_level\_set(swigCPtr, this, IntVector.getCPtr(value), value);

}

public IntVector getCum\_nneighbor\_per\_level() {

long cPtr = swigfaissJNI.HNSW\_cum\_nneighbor\_per\_level\_get(swigCPtr, this);

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

}

public void setLevels(IntVector value) {

swigfaissJNI.HNSW\_levels\_set(swigCPtr, this, IntVector.getCPtr(value), value);

}

public IntVector getLevels() {

long cPtr = swigfaissJNI.HNSW\_levels\_get(swigCPtr, this);

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

}

public void setOffsets(Uint64Vector value) {

swigfaissJNI.HNSW\_offsets\_set(swigCPtr, this, Uint64Vector.getCPtr(value), value);

}

public Uint64Vector getOffsets() {

long cPtr = swigfaissJNI.HNSW\_offsets\_get(swigCPtr, this);

return (cPtr == 0) ? null : new Uint64Vector(cPtr, false);

}

public void setNeighbors(IntVector value) {

swigfaissJNI.HNSW\_neighbors\_set(swigCPtr, this, IntVector.getCPtr(value), value);

}

public IntVector getNeighbors() {

long cPtr = swigfaissJNI.HNSW\_neighbors\_get(swigCPtr, this);

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

}

public void setEntry\_point(int value) {

swigfaissJNI.HNSW\_entry\_point\_set(swigCPtr, this, value);

}

public int getEntry\_point() {

return swigfaissJNI.HNSW\_entry\_point\_get(swigCPtr, this);

}

public void setRng(SWIGTYPE\_p\_faiss\_\_RandomGenerator value) {

swigfaissJNI.HNSW\_rng\_set(swigCPtr, this, SWIGTYPE\_p\_faiss\_\_RandomGenerator.getCPtr(value));

}

public SWIGTYPE\_p\_faiss\_\_RandomGenerator getRng() {

long cPtr = swigfaissJNI.HNSW\_rng\_get(swigCPtr, this);

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

}

public void setMax\_level(int value) {

swigfaissJNI.HNSW\_max\_level\_set(swigCPtr, this, value);

}

public int getMax\_level() {

return swigfaissJNI.HNSW\_max\_level\_get(swigCPtr, this);

}

public void setEfConstruction(int value) {

swigfaissJNI.HNSW\_efConstruction\_set(swigCPtr, this, value);

}

public int getEfConstruction() {

return swigfaissJNI.HNSW\_efConstruction\_get(swigCPtr, this);

}

public void setEfSearch(int value) {

swigfaissJNI.HNSW\_efSearch\_set(swigCPtr, this, value);

}

public int getEfSearch() {

return swigfaissJNI.HNSW\_efSearch\_get(swigCPtr, this);

}

public void setCheck\_relative\_distance(boolean value) {

swigfaissJNI.HNSW\_check\_relative\_distance\_set(swigCPtr, this, value);

}

public boolean getCheck\_relative\_distance() {

return swigfaissJNI.HNSW\_check\_relative\_distance\_get(swigCPtr, this);

}

public void setUpper\_beam(int value) {

swigfaissJNI.HNSW\_upper\_beam\_set(swigCPtr, this, value);

}

public int getUpper\_beam() {

return swigfaissJNI.HNSW\_upper\_beam\_get(swigCPtr, this);

}

public void setSearch\_bounded\_queue(boolean value) {

swigfaissJNI.HNSW\_search\_bounded\_queue\_set(swigCPtr, this, value);

}

public boolean getSearch\_bounded\_queue() {

return swigfaissJNI.HNSW\_search\_bounded\_queue\_get(swigCPtr, this);

}

public void set\_default\_probas(int M, float levelMult) {

swigfaissJNI.HNSW\_set\_default\_probas(swigCPtr, this, M, levelMult);

}

public void set\_nb\_neighbors(int level\_no, int n) {

swigfaissJNI.HNSW\_set\_nb\_neighbors(swigCPtr, this, level\_no, n);

}

public int nb\_neighbors(int layer\_no) {

return swigfaissJNI.HNSW\_nb\_neighbors(swigCPtr, this, layer\_no);

}

public int cum\_nb\_neighbors(int layer\_no) {

return swigfaissJNI.HNSW\_cum\_nb\_neighbors(swigCPtr, this, layer\_no);

}

public void neighbor\_range(long no, int layer\_no, SWIGTYPE\_p\_unsigned\_long begin, SWIGTYPE\_p\_unsigned\_long end) {

swigfaissJNI.HNSW\_neighbor\_range(swigCPtr, this, no, layer\_no, SWIGTYPE\_p\_unsigned\_long.getCPtr(begin), SWIGTYPE\_p\_unsigned\_long.getCPtr(end));

}

public HNSW(int M) {

this(swigfaissJNI.new\_HNSW\_\_SWIG\_0(M), true);

}

public HNSW() {

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

}

public int random\_level() {

return swigfaissJNI.HNSW\_random\_level(swigCPtr, this);

}

public void fill\_with\_random\_links(long n) {

swigfaissJNI.HNSW\_fill\_with\_random\_links(swigCPtr, this, n);

}

public void add\_links\_starting\_from(DistanceComputer ptdis, int pt\_id, int nearest, float d\_nearest, int level, SWIGTYPE\_p\_omp\_lock\_t locks, VisitedTable vt) {

swigfaissJNI.HNSW\_add\_links\_starting\_from(swigCPtr, this, DistanceComputer.getCPtr(ptdis), ptdis, pt\_id, nearest, d\_nearest, level, SWIGTYPE\_p\_omp\_lock\_t.getCPtr(locks), VisitedTable.getCPtr(vt), vt);

}

public void add\_with\_locks(DistanceComputer ptdis, int pt\_level, int pt\_id, SWIGTYPE\_p\_std\_\_vectorT\_omp\_lock\_t\_t locks, VisitedTable vt) {

swigfaissJNI.HNSW\_add\_with\_locks(swigCPtr, this, DistanceComputer.getCPtr(ptdis), ptdis, pt\_level, pt\_id, SWIGTYPE\_p\_std\_\_vectorT\_omp\_lock\_t\_t.getCPtr(locks), VisitedTable.getCPtr(vt), vt);

}

public int search\_from\_candidates(DistanceComputer qdis, int k, LongVector I, SWIGTYPE\_p\_float D, HNSW.MinimaxHeap candidates, VisitedTable vt, HNSWStats stats, int level, int nres\_in) {

return swigfaissJNI.HNSW\_search\_from\_candidates\_\_SWIG\_0(swigCPtr, this, DistanceComputer.getCPtr(qdis), qdis, k, SWIGTYPE\_p\_long\_long.getCPtr(I.data()), I, SWIGTYPE\_p\_float.getCPtr(D), HNSW.MinimaxHeap.getCPtr(candidates), candidates, VisitedTable.getCPtr(vt), vt, HNSWStats.getCPtr(stats), stats, level, nres\_in);

}

public int search\_from\_candidates(DistanceComputer qdis, int k, LongVector I, SWIGTYPE\_p\_float D, HNSW.MinimaxHeap candidates, VisitedTable vt, HNSWStats stats, int level) {

return swigfaissJNI.HNSW\_search\_from\_candidates\_\_SWIG\_1(swigCPtr, this, DistanceComputer.getCPtr(qdis), qdis, k, SWIGTYPE\_p\_long\_long.getCPtr(I.data()), I, SWIGTYPE\_p\_float.getCPtr(D), HNSW.MinimaxHeap.getCPtr(candidates), candidates, VisitedTable.getCPtr(vt), vt, HNSWStats.getCPtr(stats), stats, level);

}

public SWIGTYPE\_p\_std\_\_priority\_queueT\_std\_\_pairT\_float\_int\_t\_t search\_from\_candidate\_unbounded(SWIGTYPE\_p\_std\_\_pairT\_float\_int\_t node, DistanceComputer qdis, int ef, VisitedTable vt, HNSWStats stats) {

return new SWIGTYPE\_p\_std\_\_priority\_queueT\_std\_\_pairT\_float\_int\_t\_t(swigfaissJNI.HNSW\_search\_from\_candidate\_unbounded(swigCPtr, this, SWIGTYPE\_p\_std\_\_pairT\_float\_int\_t.getCPtr(node), DistanceComputer.getCPtr(qdis), qdis, ef, VisitedTable.getCPtr(vt), vt, HNSWStats.getCPtr(stats), stats), true);

}

public HNSWStats search(DistanceComputer qdis, int k, LongVector I, SWIGTYPE\_p\_float D, VisitedTable vt) {

return new HNSWStats(swigfaissJNI.HNSW\_search(swigCPtr, this, DistanceComputer.getCPtr(qdis), qdis, k, SWIGTYPE\_p\_long\_long.getCPtr(I.data()), I, SWIGTYPE\_p\_float.getCPtr(D), VisitedTable.getCPtr(vt), vt), true);

}

public void reset() {

swigfaissJNI.HNSW\_reset(swigCPtr, this);

}

public void clear\_neighbor\_tables(int level) {

swigfaissJNI.HNSW\_clear\_neighbor\_tables(swigCPtr, this, level);

}

public void print\_neighbor\_stats(int level) {

swigfaissJNI.HNSW\_print\_neighbor\_stats(swigCPtr, this, level);

}

public int prepare\_level\_tab(long n, boolean preset\_levels) {

return swigfaissJNI.HNSW\_prepare\_level\_tab\_\_SWIG\_0(swigCPtr, this, n, preset\_levels);

}

public int prepare\_level\_tab(long n) {

return swigfaissJNI.HNSW\_prepare\_level\_tab\_\_SWIG\_1(swigCPtr, this, n);

}

public static void shrink\_neighbor\_list(DistanceComputer qdis, SWIGTYPE\_p\_std\_\_priority\_queueT\_faiss\_\_HNSW\_\_NodeDistFarther\_t input, SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_HNSW\_\_NodeDistFarther\_t output, int max\_size) {

swigfaissJNI.HNSW\_shrink\_neighbor\_list(DistanceComputer.getCPtr(qdis), qdis, SWIGTYPE\_p\_std\_\_priority\_queueT\_faiss\_\_HNSW\_\_NodeDistFarther\_t.getCPtr(input), SWIGTYPE\_p\_std\_\_vectorT\_faiss\_\_HNSW\_\_NodeDistFarther\_t.getCPtr(output), max\_size);

}

}