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

\* 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 IndexLSH extends IndexFlatCodes {

private transient long swigCPtr;

protected IndexLSH(long cPtr, boolean cMemoryOwn) {

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

swigCPtr = cPtr;

}

protected static long getCPtr(IndexLSH 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\_IndexLSH(swigCPtr);

}

swigCPtr = 0;

}

super.delete();

}

public void setNbits(int value) {

swigfaissJNI.IndexLSH\_nbits\_set(swigCPtr, this, value);

}

public int getNbits() {

return swigfaissJNI.IndexLSH\_nbits\_get(swigCPtr, this);

}

public void setRotate\_data(boolean value) {

swigfaissJNI.IndexLSH\_rotate\_data\_set(swigCPtr, this, value);

}

public boolean getRotate\_data() {

return swigfaissJNI.IndexLSH\_rotate\_data\_get(swigCPtr, this);

}

public void setTrain\_thresholds(boolean value) {

swigfaissJNI.IndexLSH\_train\_thresholds\_set(swigCPtr, this, value);

}

public boolean getTrain\_thresholds() {

return swigfaissJNI.IndexLSH\_train\_thresholds\_get(swigCPtr, this);

}

public void setRrot(RandomRotationMatrix value) {

swigfaissJNI.IndexLSH\_rrot\_set(swigCPtr, this, RandomRotationMatrix.getCPtr(value), value);

}

public RandomRotationMatrix getRrot() {

long cPtr = swigfaissJNI.IndexLSH\_rrot\_get(swigCPtr, this);

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

}

public void setThresholds(FloatVector value) {

swigfaissJNI.IndexLSH\_thresholds\_set(swigCPtr, this, FloatVector.getCPtr(value), value);

}

public FloatVector getThresholds() {

long cPtr = swigfaissJNI.IndexLSH\_thresholds\_get(swigCPtr, this);

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

}

public IndexLSH(long d, int nbits, boolean rotate\_data, boolean train\_thresholds) {

this(swigfaissJNI.new\_IndexLSH\_\_SWIG\_0(d, nbits, rotate\_data, train\_thresholds), true);

}

public IndexLSH(long d, int nbits, boolean rotate\_data) {

this(swigfaissJNI.new\_IndexLSH\_\_SWIG\_1(d, nbits, rotate\_data), true);

}

public IndexLSH(long d, int nbits) {

this(swigfaissJNI.new\_IndexLSH\_\_SWIG\_2(d, nbits), true);

}

public SWIGTYPE\_p\_float apply\_preprocess(long n, SWIGTYPE\_p\_float x) {

long cPtr = swigfaissJNI.IndexLSH\_apply\_preprocess(swigCPtr, this, n, SWIGTYPE\_p\_float.getCPtr(x));

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

}

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

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

}

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

swigfaissJNI.IndexLSH\_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 transfer\_thresholds(LinearTransform vt) {

swigfaissJNI.IndexLSH\_transfer\_thresholds(swigCPtr, this, LinearTransform.getCPtr(vt), vt);

}

public IndexLSH() {

this(swigfaissJNI.new\_IndexLSH\_\_SWIG\_3(), true);

}

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

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

}

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

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

}

}