package com.twitter.search.core.earlybird.index.util;

import com.google.common.base.Preconditions;

public abstract class SearchSortUtils {

public interface Comparator<T> {

/\*\*

\* Compares the item represented by the given index with the provided value.

\*/

int compare(int index, T value);

}

/\*\*

\* Performs a binary search using the given comparator, and returns the index of the item that

\* was found. If foundLow is true, the greatest item that's lower than the provided key

\* is returned. Otherwise, the lowest item that's greater than the provided key is returned.

\*/

public static <T> int binarySearch(Comparator<T> comparator, final int begin, final int end,

final T key, boolean findLow) {

int low = begin;

int high = end;

Preconditions.checkState(comparator.compare(low, key) <= comparator.compare(high, key));

while (low <= high) {

int mid = (low + high) >>> 1;

int result = comparator.compare(mid, key);

if (result < 0) {

low = mid + 1;

} else if (result > 0) {

high = mid - 1;

} else {

return mid;

} // key found

}

assert low > high;

if (findLow) {

return high < begin ? begin : high;

} else {

return low > end ? end : low;

}

}

}