/\*\*

A quick sort algorithm that uses iteration to

sort an array from low to high numbers.

@author Pj Kim

\*\*/

import java.util.Random;

public class QuickIterative

{

public static void main(String[] args) throws EmptyQueueException //need exception for empty queues

{

LinkedQueue<T> queue = new LinkedQueue<T>(); //Creates new linked queue

int[] array = new int[10];

Random randomObj = new Random();

//fills each index of the array with random numbers.

for (int i = 0; i < array.length; i++)

{

array[i] = randomObj.nextInt(1000);

System.out.print(array[i] + " ");

}

System.out.println();

//Define initial pointer positions and variables

int leftPointer = 0;

int rightPointer = array.length - 1;

boolean finishedSorting = false; //will be true after running

int pivotPoint;

int pivot;

int index;

while (!finishedSorting) //loops if more sorting is required

{

pivot = array[rightPointer]; //rearrange pivot and index

index = leftPointer - 1;

for (int i = leftPointer; i < rightPointer; i++)

{

if (array[i] <= pivot) //swap values if less than the pivot.

{

index++;

int temp = array[index];

array[index] = array[i];

array[i] = temp;

}

}

//swap first value greater than pivot with the pivot.

pivotPoint = index + 1;

array[rightPointer] = array[index + 1];

array[index + 1] = pivot;

queue.enqueue(leftPointer); //adds left partition to queue

queue.enqueue(pivotPoint - 1);

queue.enqueue(pivotPoint + 1); //adds right partition queue

queue.enqueue(rightPointer);

do

{

if (queue.isEmpty()) //When queue is empty, nothing left to sort

{

finishedSorting = true;

}

else //when queue is not empty, more sorting to do

{

leftPointer = queue.dequeue();

rightPointer = queue.dequeue();

}

}

while (leftPointer >= rightPointer && !finishedSorting);

//if leftPointer >= rightPointer, this segment of the array does not need to be sorted anymore, so next two pointers should be pulled from the queue

}

//prints out the sorted array

for (int j = 0; j < array.length; j++)

{

System.out.print(array[j] + " ");

}

}

}