CSCI 321 Computer Science III Summer 2019

Lecture 4 Activity 1

1. Write a comparator for nonnegative integers that determines order based on the number of 1’s in each integer’s binary expansion, so that i < j if the number of 1’s in the binary representation of i is less than the number of 1’s in the binary representation of j. Write a test suite showing an example of a priority queue sorting by the number of 1’s in each integer’s binary representation.

Hint: Write a method that computes the number of 1’s in the binary expansion of an integer by using the bitwise “and” operation. See the code of an example of a priority queue sorting by string length as follows.

// Test.java

import java.util.Comparator;

import java.util.PriorityQueue;

public class Test

{

public static void main(String[] args)

{

Comparator<String> comparator = new StringLengthComparator();

PriorityQueue<String> queue =

new PriorityQueue<String>(10, comparator);

queue.add("short");

queue.add("very long indeed");

queue.add("medium");

while (queue.size() != 0)

{

System.out.println(queue.remove());

}

}

}

// StringLengthComparator.java

import java.util.Comparator;

public class StringLengthComparator implements Comparator<String>

{

@Override

public int compare(String x, String y)

{

// Assume neither string is null. Real code should

// probably be more robust

// You could also just return x.length() - y.length(),

// which would be more efficient.

if (x.length() < y.length())

{

return -1;

}

if (x.length() > y.length())

{

return 1;

}

return 0;

}

}

Output:

short

medium

very long indeed

The following links might be helpful.

Priority Queue in Java: <https://docs.oracle.com/javase/7/docs/api/java/util/PriorityQueue.html>

Bitwise and Bit shift operators in Java:

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op3.html