**Com S 311 Spring 2021**

# Assignment 2: Implementing an Extended Priority Interface

**Due: March 12, 11:59 pm**
**Early Submission: March 11, 11:59 p.m. (5% bonus)**
**Late Submission: March 13, 11:59 A.M. (25% penalty)**

## Introduction

This assignment gives you an opportunity to work on the heap data structure by implementing an extended priority interface in Java. The interface is defined in a given template code segment, with many of its methods already completed. Your task is to implement the remaining methods marked by TODO before their method headings. You should study the completed methods along with comments carefully before you complete the remaining methods. Note that Java is 0-based for array and list indexes, instead of 1-based. If an array $A[0..n-1]$ of length $n$ is used to implement a nearly complete tree for a heap, then the index for the parent of a child at index $j$ is $\lfloor j-1 \rfloor/2$, the index for the left child of a parent at index $j$ is $2j+1$, and that for the right child is $2j+2$. In the template code segment, an instance of ArrayList<E> is used as an array list for the heap, where indexes in this array list of length $n$ are $0, 1, ..., n-1$. Note that if an element is removed from the heap, then an element is also removed from this array list so that the length of this array list is always the size of the heap.

You are allowed to discuss with classmates, but you must work on the homework problems on your own. You should write the final code alone, without consulting anyone.

Below is a sample code segment to use the Heap class along with its output.

```
public static void main(String[] args)
{
   Heap<Integer>  pq = new Heap<Integer>();
   pq.add(10);
   pq.add(15);
   pq.add(20);
   pq.add(30);
   pq.add(25);
   pq.add(25);
   pq.add(30);
   pq.add(40);
   pq.add(35);
   pq.add(50);
```

```
     pq.add(10);
     pq.showHeap();
     System.out.println( pq.getLastInternal() );
     pq.trimEveryLeaf();
     pq.showHeap();
     while ( ! pq.isEmpty() )
     {
       System.out.println( pq.removeMin() );
     }

     List<String>  alist = new ArrayList<String>();
     alist.add("TGA");
     alist.add("ACG");
     alist.add("GCT");
     alist.add("GTA");
     System.out.println( "Before sorting: " + alist.toString() );
     heapSort(alist);
     System.out.println( "After sorting: " + alist.toString() );
  }
} // Heap

// The root is shown on the leftmost column, with its children on the next column,
// The string null means no child is present.
Output:

>10
 >10
  >30
   >40
    >null
    >null
   >35
    >null
    >null
  >15
   >50
    >null
    >null
   >25
    >null
    >null
 >20
```

```
  >25
   >null
   >null
  >30
   >null
   >null
15
>10
 >10
  >30
   >null
   >null
  >15
   >null
   >null
 >20
  >null
  >null
10
10
15
20
30
Before sorting: [TGA, ACG, GCT, GTA]
After sorting: [ACG, GCT, GTA, TGA]
```

## Submission

You are required to include, in your submission, the source code for each of the classes: THe template code segment is given in package cs311hw2.zip.

Write your class so that its package name is edu.iastate.cs311.hw2. Your source files (.java files) will be placed in the directory edu/iastate/cs311/hw2 (Linux) or edu\iastate\cs311\hw2 (Windows), as defined by the package specified above. Be sure to put down your name after the @author tag in each class source file. Your zip file should be named Firstname_Lastname_HW2.zip. You may submit a draft version of your code early to see if you have any submission problem on Canvas. We will grade only your latest submission.