Homework 6

Please submit your solution via email to the instructor with CC to yjk337@nyu.edu.

The deadline for Homework 6 is March 27, 1 pm.

Problem 1  Faulty Heap Implementation (10 Points)

Consider the following (faulty) implementation of the removeFirst operation into a binary heap data structure:

```java
/*@
public normal_behavior
@  requires !isEmpty();
@  ensures \old(queue).has(\result) &&
   \old(queue).equals(\old(queue).remove(\result)) &&
   \forall Comparable o; queue.has(o);\forall \result.compareTo(o) <= 0);
@  modifies queue;
@*/
public Comparable removeFirst() {
    Comparable first = elems[0];
    Comparable last = elems[--numElems];
    int pos = 0;
    int child = 1;
    while (child < numElems) {
        if (child + 1 < numElems &&
            elems[child].compareTo(elems[child+1]) < 0) {
            child++;
        }
        if (elems[child].compareTo(last) < 0)
            break;
        elems[pos] = elems[child];
        pos = child;
        child = 2*pos;
    }
    elems[pos] = last;
    //@ set ghostQueue = ghostQueue.remove(first);
    return first;
}
```

The source code for this faulty implementation with additional JML specifications is provided on the course webpage. Use JMLUnit to generate test cases that reveal the error(s) in the implementation. Then change the implementation such that the specification is met.
Problem 2  Faulty Insertion Sort (15 Points)

Consider the following (faulty) implementation of insertion sort.

```java
public class InsertionSort {
    public static void sort(int[] arr) {
        for (int i = 1; i <= arr.length; i++) {
            for (int j = i; j >= 0; j--) {
                if (arr[j] >= arr[j-1]) {
                    int tmp = arr[j];
                    arr[j] = arr[j-1];
                    arr[j] = tmp;
                }
            }
        }
    }
}
```

Write an appropriate specification for method `sort` and generate test cases using JMLUnit to reveal the error(s) in the implementation. Then change the implementation such that your specification is met. *Hint:* you can use JML’s `\forall` construct to specify a post-condition stating that the array is sorted.