Lab Report 09: Min Heap of Integers

# Problem

Construct a min heap type which supports the operations add, pop, peek, and heapsort. The heap should be in min order; that is, lowest elements should be popped first.

# Proposed Solution

Create a generic min heap class which uses an array to store objects. Use the knowledge that the child of a node will be at the (index \* 2 + 1) node to move from parent to children and vice-versa. Create a copy of the heap which pops into a newly created array in order to implement heapsort.

# Tests and Results PS C:\Users\jynelson\Documents\146\Labs\09> java test.MinHeapTester Min heap tester Populating with values 21 37 49 11 23 1 13 16 33 17 Printing heap 1 16 11 21 17 49 13 37 33 23 Testing heap sort 1 11 13 16 17 21 23 33 37 49 Removing top item (1) from heap 11 16 13 21 17 49 23 37 33

# Problems Encountered

Java does not support copying from one generic array to another. I wrote in more detail of the problems and solutions at [this link](https://jyn514.github.io/2018/03/29/Copying-Generic-Arrays-in-Java.html); essentially there has to be an instance of the generic type in order to create a type-safe array; otherwise the JIT compiler will throw a ClassCastException.

# Conclusions and Discussion

In the future, I would combine min- and max-heaps into a single parameterized class; the code for both is extremely similar and it would take much less time to create if it were a single class.

Overall I found this lab interesting and challenging.

# Additional Questions

1. Demonstrate each step of inserting the elements 21, 27, 49, 11, 23, 1, 13, 16, 33, and 17 into this min heap.

2. Demonstrate each step of inserting the same elements in a max heap.

3. Show each step when deleting two elements from the min heap.