Programming Assignment #1

**Programmer:** Timothy Vicars

**Course:** CSCI 3352

**Date:** Oct 1, 2019

**Programming Assignment:** #1 – Heapsort and Quicksort

**Environment:** Microsoft Windows Visual C++, Linux GCC C++

**Objective:** This programming assignment was to implement quicksort and Heapsort Algorithms

**Algorithm:**

**procedure** heapsort(a, count) **is**

**input:** an unordered array *a* of length *count*

heapify(a, count)

end ← count - 1

**while** end > 0 **do**

*(a[0] is the root and largest value. The swap moves it in front of the sorted elements.)*

swap(a[end], a[0])

*(the heap size is reduced by one)*

end ← end - 1

*(the swap ruined the heap property, so restore it)*

siftDown(a, 0, end)

quickSort(arr[], low, high)

{

if (low < high)

{

/\* pi is partitioning index, arr[pi] is now

at right place \*/

pi = partition(arr, low, high);

quickSort(arr, low, pi - 1); // Before pi

quickSort(arr, pi + 1, high); // After pi

}

}

**Input**: An array of numbers

**Output**: a sorted array of numbers

**Compile:** (linux):

$ g++ -o quicksort quicksort.cpp

$ g++ -o heapsort heapsort.cpp

**Run**: ./quicksort

./heapsort

