Due: Tuesday, November 2

Historical background. In 1964, J. W. J. Williams published an article in *Communications of the ACM* introducing a new sorting algorithm called Heapsort. The algorithm relies on a data structure of independent interest also invented by Williams, a binary heap. He described a method of heap construction that an easy analysis shows to run in time $O(n \log n)$. Later that same year, Robert Floyd[1] showed that a heap can actually be constructed in $O(n)$ time. We will refer to these two constructions as the Williams and Floyd constructions. The latter is also commonly referred to as the *bottom-up* construction.

By the way, the $O(n \log n)$ performance of the Williams construction is a worst-case measure. It can be shown to actually run in linear time on average[2]; even so, the Floyd construction tends to be faster.

Purpose. The purpose of this assignment is to compare the running times of these two heap construction methods on randomly generated inputs. We will use the number of swaps as a proxy for the running time.

To begin, you must create a file *heaps.cpp* that defines the following two functions:

```
// Transforms list into a heap using the Williams method.
// Returns the number of swaps performed.
int make_heap_williams(int list[], int size);

// Transforms list into a heap using the Floyd (bottom-up) method.
// Returns the number of swaps performed.
int make_heap_floyd(int list[], int size);
```

For swapping array elements, use the STL swap function.

I will test your code with a program that calls your functions on arrays of different sizes and verifies that the resulting array in each case is in fact a heap. You will of course want to test your work in the same way. My test program *must* compile with your *heaps.cpp*, so you cannot change the signature or return type of either function.

Next, write a program called *compare_heaps.cpp* that #includes your implementation *heaps.cpp*. The program reads two integers *n* and *k* at the command line, and performs the following experiment *k* times:

- Create an array of size *n* initialized with random integers.
- Create a copy of the first array.
- Use *make_heap_williams* to transform the first array into a heap.
- Use *make_heap_floyd* to transform the second array into a heap.

Output the average number of swaps over all *k* trials performed by each of the two heap-building functions. Does one consistently outperform the other? To address this question, experiment with a wide range of values for *n* and the largest value of *k* that your program can handle without running too slow. Draw a conclusion about relative performance and include some of your experimental results to support it in a text file named *results.txt*.

What to submit: *heaps.cpp*, *compare_heaps.cpp*, and *results.txt*.

---

[1] Turing Award winner (1969). He also published what is now known as the Floyd-Warshall algorithm, which was independently published in different form by earlier authors. It is a DP (dynamic programming) graph algorithm described in Chapter 13.4.2 of our book.

[2] Averaging over all possible permutations of input {1, 2, …, *n*}.