Andrew Hsu

LAB0

I chose the mergesort algorithm because it is most familiar to me. I have done several recursive sorting algorithms in the past but mergesort has been my go-to because the splitting of arrays makes sense to me.

My code is split into three functions: main(), mergesort(), and merge(). Main is responsible for handling the input text file and calling mergesort and printing the sorted array. Mergesort is a function that splits an input array into two and calls merge afterwards. The function merge() is the function that does all the heavy lifting. Taking an input of two arrays, it compares two values inside the two arrays and puts them in ascending order in the new output array. Main calls merge sort, which is a recursive function, and mergesort calls merge.

Sort.cpp is the main c file that contains all working functions. Makefile is the makefile run using command “make”. Mysort is the output file that is run using ./sort <input text file>. Test.txt is a randomly generated file using shuf. And Writeup.doc is the name of this writeup file.

To compile, just run command “make” inside the directory.

To run, enter command ./mysort <inputfile.txt> and the sorted array should print to the console.

There are no bugs that I have encountered so far. However I have not tested an extremely large array or different formats of files. I also have not checked for malformed inputs inside the txt file.