

Assignment 2: Fun with Sorting!

1. **In groups of 3**, write two programs in C which perform sorting yet use different algorithms. Each program prompts the user to enter an integer n and an input file containing a list of 800000 numbers. Each program then reads in the first n entries found in the specified input file and prints the sorted list to a file.

The two programs implement this functionality in the following ways:

- a. (5 marks) The first program implements in-place insertion sort.
 - b. (7 marks) The second program implements in-place heap sort (*as outlined in the Intro. to Sorting set of slides*).
2. (4 marks) Run your programs on the input sizes (a) 200,000, (b) 400,000 (c) 600,000 and (d) 800,000 using the three input files provided.

For each value of n specified above, run your programs five times. Record only the time it takes to sort all n numbers -- *not the time to read in the numbers from the file, nor the time to print the sorted list to a file*. Include all timings in your report but summarize your results by providing a table like that shown below.

Input Size	Average time for 5 runs of In-place Insertion Sort			Average time for 5 runs of In-place Heap Sort		
	AlmostSorted.txt	Random.txt	Reverse.txt	AlmostSorted.txt	Random.txt	Reverse.txt
$n = 200000$						
$n = 400000$						
$n = 600000$						
$n = 800000$						

3. (4 marks) Given the experimental data obtained in #2, calculate the rate of growth of time with respect to the size of the input of each algorithm per input file and express using *Big-Oh notation*. Provide a discussion as to why the timings were or were not what you expected for a given input file.

To measure execution time in C:

```
/* The following header file defines clock_t datatype, clock() function, as well
   as the constant CLOCKS_PER_SEC */
#include <time.h>

clock_t start = clock();

/* Code you want timed here */

double timeElapsed = ((double)clock() - start) / CLOCKS_PER_SEC;
```

To read from a file:

```
FILE *fp;
fp = fopen("inputFile.txt", "r");
if (fp == NULL) {
    printf("Error opening inputFile.txt");
    exit(1);
}

//create array size according to user input and read values into array
int array[someValue];

int i;

for (i = 0; i < someValue; i++)
{
    fscanf(fp, "%d", &array[i]);
}
```