**Assignment 1**

**3.Timing: Part 1 (20 Points):**

Compile and run the program without any extra optimizations, but with *profiling* for timing:

gcc -c -pg -O0 main.c

gcc -c -pg -O0 mergeSort.c

gcc -c -pg -O0 insertionSort.c

gcc main.o mergeSort.o insertionSort.o -pg -O0 -o assign1-0

*Run the program twice* timing it both times, and answer the following:

* 1. *How many****self seconds****did insertionSort() take?*

It took 14.96 seconds.

* 1. *How many****self seconds****did mergeSort() take?*

It took .02 seconds.

**4.Timing: Part 2 (20 Points):**

Compile and run the program *with* optimization, but with *profiling* for timing:

gcc -c -pg -O2 main.c

gcc -c -pg -O2 mergeSort.c

gcc -c -pg -O2 insertionSort.c

gcc main.o mergeSort.o insertionSort.o -pg -O2 -o assign1-2

*Run the program twice* timing it both times, and answer the following:

* 1. *How many****self seconds****did insertionSort() take?*

It took 11.13 seconds.

* 1. *How many****self seconds****did mergeSort() take?*

It took 0.02 seconds.

1. **Human vs. Compiler Optimization (10 Points):**

Which is faster:

* 1. A bad algorithm and data-structure optimized with -O2
  2. A good algorithm and data-structure optimized with -O0

Explain your answer.

I think a good algorithm with -o0 optimization is faster in way that our algorithm is made to be more efficient and is less expensive for the user.

1. **Parts of an executable (Points 20):**

Please find the following inside of assign1-0 by using objdump.

* 1. If it *can* be found then *both*
     1. Give the objdump command, and
     2. Show the objdump result
  2. If it *cannot* be found then tell why not. Where in the memory of the runtime process is it?

Look for:

1. The string constant in main()
2. Global integer numNumbers in main.c
3. The code for freeList()
4. The pointer argument nodePtr in freeList()

|  |  |  |
| --- | --- | --- |
| **Question** | **Command** | **Result** |
| (A) | objdump -s -j .rodata assign1-0 | It shows the following:  (………%d……………  How do you want to sort %d numbers?  (1) Insertion sort.  (2) Merge sort.  Your choice (1 or 2)?)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (B) | objdump -s -j .data assign1-0\_\_\_\_\_ | 607020 00000000 000001000 ……  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (C) | objdump -d -j .text assign1-0 | It starts at the address of 4000902<freeList>:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| D | objdump -d -j .text assign1-0 | Cant be found. |