CS361 Programming Project: Sorting

Due: Dec 6th (Wednesday) End of Day

Sorting is at the core of a lot of algorithms. The goal of this project is to implement three common sorting algorithms and compare their performances.

1. Implementation:

You are allowed to work in groups of **up to 4** students.

The algorithms:

(1) Sorting using a quad heap: You will be implementing a modified heap called a quad-heap, which is a degree 4 tree that is filled on all levels except the very last, that is filled from left to right.

(2) 3-way merge sort: You will be implementing a 3-way merge sort, which divides the given input into 3 subsets of roughly equal size and recursively sort each subset.

(3) Randomized quick sort: You will implement the randomized quick sort algorithm. In each iteration, the pivot must be randomly chosen using a random number generator.

You may choose to use C, C++, Java or python. However, all algorithms must be implemented using the **same** programming language.

You may **NOT** use any data structures from existing libraries except: (1) basic data structures such as arrays, (2) a random number generator, (3) system functions, such as print to screen, write to file, etc.

2. Performance benchmarking:

(1) For each of the algorithm, you will benchmark its running time with respect to input data of size: $2^{20}, 2^{21}, 2^{22}, 2^{23}, 2^{24}, 2^{25}, 2^{26}, 2^{27}, 2^{28}, 2^{29}, 2^{30}$. You should exclude the I/O time when benchmarking. In other words, if your program is reading in the input from a file, your timer should start after all the data is loaded into the main memory.

(2) When benchmarking, be sure to also compare the running time of these three algorithms when inputs are integers vs. double precision floating point numbers.

(3) Compare the running times between the three algorithms.

3. Project Report:

You will need to submit a final report of your findings. The report must be typewritten in a scientific paper format with a minimum 11pt font size. The report should include the following key sections:

(1) Title, authors, and institutions of the authors.

(2) Introduction.

(3) Preliminaries: You will introduce the key ideas and performance characteristics of the implemented algorithms, and describe the algorithms exactly as implemented. As an example, in the preliminaries, you should discuss how the parent-child relation is calculated in a quad-heap.

(4) Performance analysis and discussions. Here are some of the things you should discuss in your report: (1) Are the asymptotic running times of these algorithms close to their actual

performance? What is the constant factor you have found in your running time analysis? (2) Do all three programs have the same coding complexity? What about debugging effort? (3) Which algorithm is the fastest or slowest? Why?

(5) Conclusion

(6) Bibliography

(7) **Contributions of Each Team Members:** describe in detail the contribution of each member of your team. For example, member A implemented the quad-heap, member B implemented quick sort, member C implemented 3-way merge sort, and member D is the lead in performing benchmarks and compiling the final report.

(8) Acknowledgement

4. Submission:

You will be uploading the following on Canvas by midnight on Dec 6th (Wednesday):

(1) your final report in a single pdf file. Only one report is needed for a team.

(2) your source code and a small testing data set in a single zipped file.

Note that there will not be an extension for the final project. Each team has almost 2 months to work on the project and there is no excuse for not completing it on time. Start now!