| COS 397 Spring 2010 | Class Exercise 1 | 16 questions; 8 pgs. |
|---------------------|------------------|----------------------|
|                     |                  |                      |

© 2010 Sudarshan S. Chawathe

This exercise is designed to complement the paper on sorting in manycore GPUs<sup>1</sup> assigned as reading, with the goal of highlighting both the technical aspects and the presentation and research-conduct aspects as they relate to Capstone projects.

- 1. List the members of your group below:
- 2. Define comparison sort.

3. Indicate the manner in which each of the first 15 references cited in the paper is related to the work described in the paper, by annotating it using one or more of: application; background information; component or building-block (conceptual or concrete); competition; other (specify).

<sup>&</sup>lt;sup>1</sup>Nadathur Satish, Mark Harris and Michael Garland, "Designing Efficient Sorting Algorithms for Manycore GPUs," in *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium* (*IPDPS*) (Rome, Italy, 2009).

4. Provide pseudocode for a radix sort algorithm and illustrate its operation on the following input.

 $39\;10\;3\;89\;24\;26\;58\;74\;26\;48$ 

5. Repeat Question 4 for merge sort.

- 6. How many concurrent threads are supported in each SM multiprocessor of currentgeneration NVIDIA GPUs? How many cores does each SM contain? What is the range of the total number of cores?
- 7. What are *warps*? What is thread divergence and why is it important?

8. True or false: If the number of concurrent threads far exceeds the number of physical compute cores, performance will suffer. Explain your answer.

9. What are coalesced writes and why are they important?

10. Explain the reasoning behind treating each thread block as a processor for the purpose of algorithm analysis.

11. What is CRCW PRAM?

12. The paper does not list any code or pseudocode. How may we study the details of the implementations described in the paper, perhaps to improve on them?

13. What is stable sorting and why is it important in radix sorting?

14. Describe the salient details of Figure 1 from the paper as precisely and concisely as possible. That is, convey the important points without using the figure. Repeat for Figure 2.

15. How does the paper's implementation of GPU radix sort take advantage of the GPU shared memory?

16. Describe sorting by 1-bit splits and illustrate its action on the data of Question 4.

[additional space for answering the earlier question]