CSE 332 Project 3 Write up

* Note: The last 3 questions require you to write code, collect data, and produce graphs of your results together with relatively long answers. Do not wait until the last minute to start this write up!

1. **Who is in your group (Your name, UW NetID)?**

2. **What assistance did you receive on this project? Include anyone or anything *except* your partner, the course staff, and the printed textbook.**

3. **a) How long did the project take?**
   b) Which parts were most difficult?
   c) How could the project be better?

4. **(OPTIONAL) What "above and beyond" projects did you implement? What was interesting or difficult about them? Describe in detail how you implemented them.**

5. **a) How did you test your program? What parts did you test in isolation and how?**
   c) What smaller inputs did you create so that you could check your answers?
   d) What boundary cases did you consider?

6. **For finding the corners of the United States and for the first grid-building step, you implemented parallel algorithms using Java's ForkJoin Framework. The code should have a sequential cut-off that can be varied. Perform experiments to determine the optimal value of this sequential cut-off.**
   1) Sequential vs. parallel versions of corner finding. Looking at V1 and V2, vary the cutoff for V2.
   2) Cut-off in the grid-building step. Looking at V3 and V4, vary grid-building cutoff for V4.
   3) Cut-off in the grid-merging step. Looking at V3 and V4, vary grid-merging cutoff for V4.
   * You don't have to worry about finding the optimal combination of cut-offs.

   a) Describe your experimental setup:
      1) Your machine characteristics
      2) How you collected timing information
      3) Any details that would be needed to replicate your experiments

b) Experimental Results: Place your graph for experiment 1), 2) and 3).
Clearly label which line is for which version in each of your plot.

1) Cutoff vs. Runtime for V1 and V2
2) Cutoff vs. Runtime for V3 and V4: Grid building
3) Cutoff vs. Runtime for V3 and V4: Grid merging

c) Interpretation of Experimental Results
Note that if the sequential cut-off is high enough to eliminate all parallelism, then you should see performance close to the sequential algorithms, but evaluate this claim empirically (and then answer the question - is this what you see?). For each of the experiments 1), 2) and 3), answer the following questions.

1) What did you expect about the result and why?
2) Did your result agree with your expectation?
3) If the result did not match with your expectation, why do you think it happened?
4) Draw a conclusion from the experimental result.

7. **Compare the performance of V4 to V5 as the size of the grid changes.**
Clearly label which line is for V4 and which line is for V5 in your plot.

a) Experimental Results: Place your graph for Grid size vs. Runtime for V4 and V5 here.
b) Intuitively, which version is better for small grids and which version is better for large grids?
c) Does the experimental data validate your hypothesis in b)? If the result did not match with your expectation, why do you think it happened?

8. **Compare the performance of V1 to V3 and V2 to V4 as the number of queries changes. That is, how many queries are necessary before the pre-processing is worth it?**
Clearly label which line is for which version in each of your plot. Note you should time the actual code answering the query, not including the time for entering the query.

a) Experimental Results (Place your graph here).
1) Number of Query vs. Runtime for V1 and V3
2) Number of Query vs. Runtime for V2 and V4

b) Interpretation of Experimental Results

For each of the experiments 1) and 2), answer the following questions:

1) What did you expect about the result and why?
2) Did your result agree with your expectation?
3) If the result did not match with your expectation, why do you think it happened?

c) According to your experiment, how many queries are necessary before the pre-processing is worth it?

9. **<u>If you worked with a partner</u>:**
   a) Describe the process you used for developing and testing your code. If you divided it, describe that. If you did everything together, describe the actual process used (eg. how long you talked about what, what order you wrote and tested, and how long it took).
   b) Describe each group member's contributions/responsibilities in the project.
   c) Describe at least one good thing and one bad thing about the process of working together.

Appendix

Place anything else that you want to add here.