CS14 – Lab 3

Warmup. The sum of the primes below 10 is 2 + 3 + 5 + 7 = 17. Find the sum of all the primes below two million. What is the run time of your algorithm? If we want to calculate the sum of all primes below two billion, approximately how long would your algorithm take?

Exercises: In this lab, you will implement Conway's game of life using a different data structure. Instead of storing the entire game in a matrix, we will store only those cells which are currently "alive" in a list.

1. Read through the code of the PositionListBoard class to understand how it works. Implement the getsquare and setsquare functions inside the PositionListBoard class. Notice that the iterate() function is derived from GameBoard, so you've already implemented that in the last lab. Once the getsquare and setsquare functions work, the entire class should work.

2. Update your main function so that you can specify to use the PositionListBoard from the command line.

3. Why is PositionListBoard::posL a list? How would the runtimes of your getsquare and setsquare functions differ if we used a vector instead?

4. What are the best and worst case run times of PositionListBoard::iterate()? How does this compare to the run time of MatrixBoard::iterate()?

5. We can create a (potentially) more efficient iterate function that takes advantage of our data structure. You will implement this iterate function in the EfficientPositionListBoard class.

   Your new iterate function will take advantage of the following fact: If a dead cell has no neighbors, there is no need to update it; it must remain dead. In the version of iterate you've already written, you checked every cell to see if it was alive or dead. Now, however, we have a list of living cells. So all we need to do is update these cells and their neighbors. All other cells must remain dead.

6. Update your main function so that you can specify to use the EfficientPositionListBoard from the command line.

7. What are the best and worst case running times of your EfficientPositionListBoard::iterate()? How does this compare to the other iterate functions?