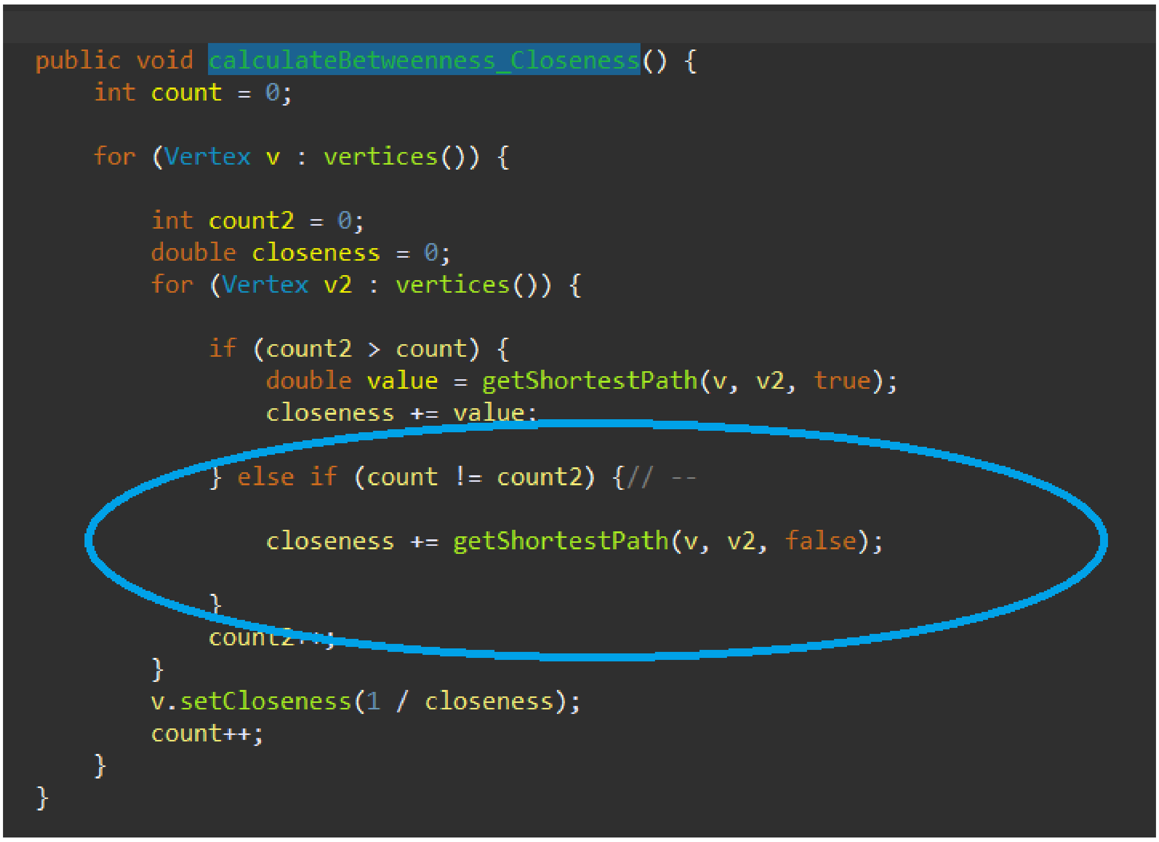
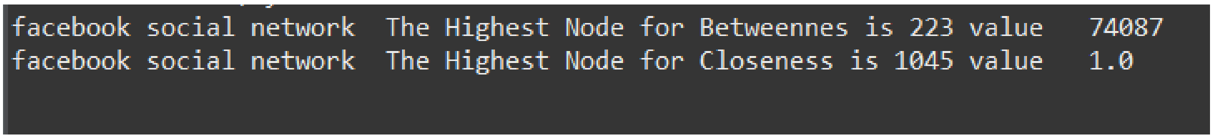
The facebook_social_network.txt file takes approximately 7 minutes to run. Most of this time is spent in the getShortestPath function. The first reason why it took a long time was that I had to put the neighbors in the arraylist while the vertexs were being formed, as the "getNeighborIterator" and "NeighborIterator" did not work properly, while navigating the neighbors with the iterator, which has the example code in the book.



```
public void calculateBetweenness_Closeness() {
    int count = 0;

    for (Vertex v : vertices()) {

        int count2 = 0;
        double closeness = 0;
        for (Vertex v2 : vertices()) {

            if (count2 > count) {
                double value = getShortestPath(v, v2, true);
                closeness += value;

            } else if (count != count2) {// --

                closeness += getShortestPath(v, v2, false);

            }
            count2++;
        }
        v.setCloseness(1 / closeness);
        count++;
    }
}
```

The second reason is that the app rarely calculates a path twice. at the location specified in the code above. Betweenness should be calculated under all circumstances, but Closeness may not be needed in some cases. For example, let a, b , c , d be vertex, the length of the path from a to b is equal to the length of the path from b to a.



```
facebook social network  The Highest Node for Betweennes is 223 value   74087
facebook social network  The Highest Node for Closeness is 1045 value   1.0
```