**CSCE 4133 – Algorithms**

**Programming Project Report**

Names:

Clayton Warstler – 010971514 – cjwarstl@uark.edu

Date:  10/27/2023

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

For this project, I was tasked with implementing binary search tree to store data in order so that we can search or find the minimum value quickly. The starter code was provided by Dr. Luu.

**Implementation:**

We started with the sample code from the HW2 review. The pseudo code was very helpful, and provided most of the info we needed. The noticeable differences between the pseudo code and our methods were that we correctly implemented a path size limit based on the number of buildings passed into our functions. Our BFS program was also a little bit different from the pseudo code provided, since we needed to implement a way to keep track of the number of shortest paths. We did this with a length and shortest path vector, which tracked and returned the number of shortest paths.

**Testing:**

All the makefile instructions as well as the compilation commands were given to us in the review. We also were given the required test cases to find the number of shortest paths, as well as the length of the shortest path for each algorithm in the code provided. This means that the only testing we had to do was run the program with the given commands. This streamlined our testing process, meaning that we could “test” the program in seconds. I am extremely grateful for this.

**Conclusions:**

Overall, we are happy with the resulting program. It works completely as expected, with the bfs returning two shortest paths, and every algorithm returns a path length of 27 as well as its corresponding path on opencv. We may end up switching partners for the next project due to scheduling issues, but we’re glad things worked out in the end for this project, as we were still able to finish in time and fix our BFS algorithm.