CS 473 Final Project Algorithm Analysis

Group Members: Andrew Hutson, McKenna Galle, Vina Le

# Empirical Analysis

Our original number of paths was 14 factorial (87,178,291,200). We increased the number of paths to 28 factorial (~3.05\*1029). The runtime for 14 factorial was 30 milliseconds (as measured by the Visual Studio 2017 debugger), and 38 milliseconds for 28 factorial (again, measured by the debugger). From this data, we calculated a ratio of ~1.267, which is an increase of ~26.7%.

# Theoretical Analysis (Worst Case)

The worst case theoretical number of paths of our branch-and-bound algorithm is n\*2n. For 14 nodes, the worst-case number of paths (assumes the correct path is found last every time) is 229,376 paths. For 28 nodes, the worst-case number of paths is 7,516,192,768 paths. This yields a ratio of 32,768, which is an increase of 3,276,700%.

# Theoretical vs. Empirical Comparison

As you can see, there is a considerable difference between the empirical data and the theoretical data. We believe this difference is due to our algorithm not running the worst case. We’re pretty sure that the program is running somewhere in-between the best-case and the worst-case efficiency.