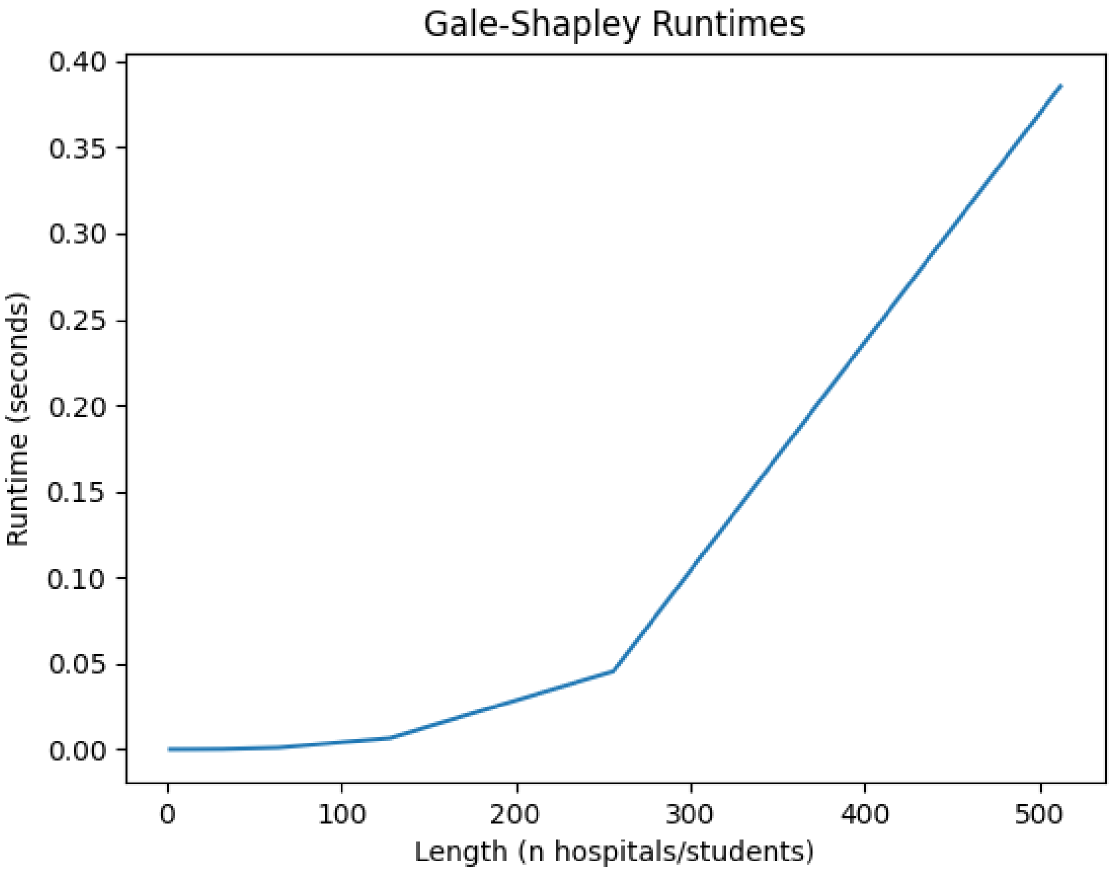
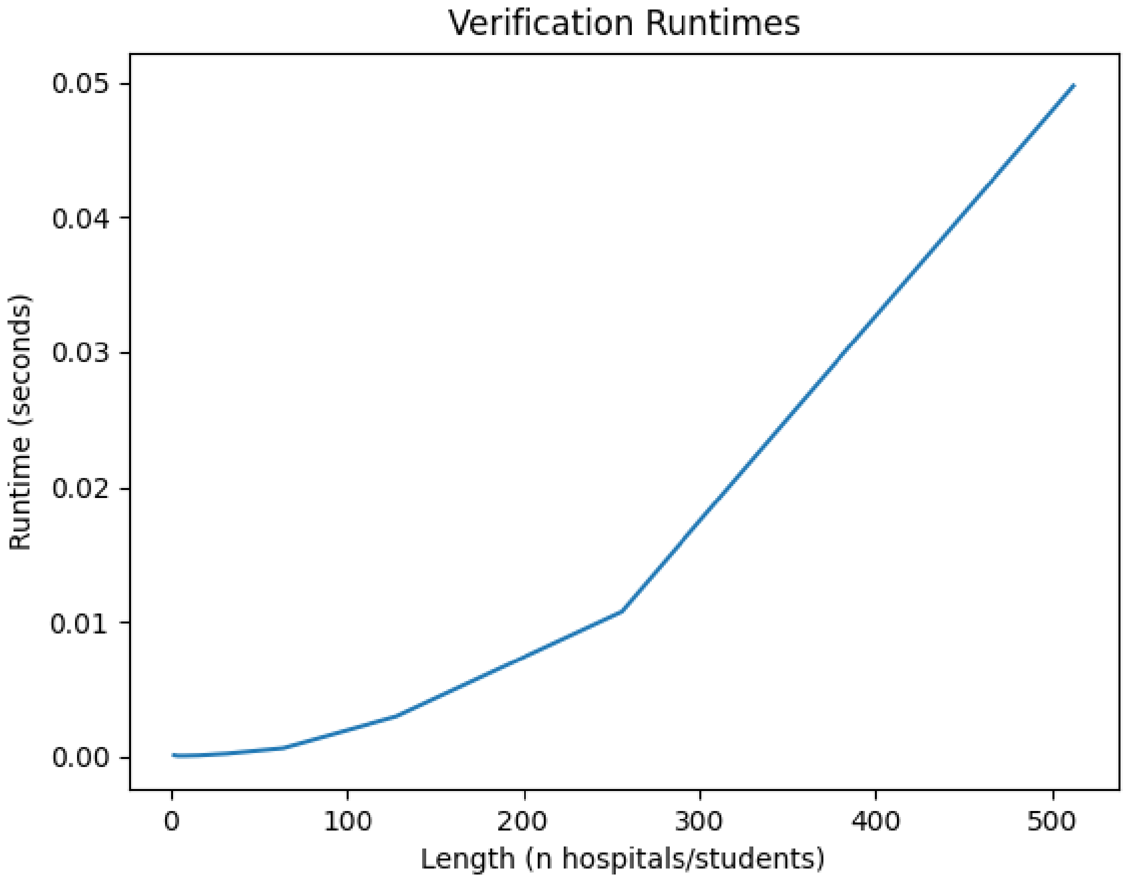


Above is the running time graph for the Gale-Shapley algorithm we implemented. The general trend of runtime is in theory n^2, and our algorithm just about matches that. It is a little worse due to slight inefficiencies in our algorithm and choice of python as programming language.

For n = 512, seconds is ~0.36, and n = 256, seconds is ~0.05. With a 2x increase in length a quadratic function will yield a 4x increase in runtime. So our algorithm is almost optimal. If we implemented some optimizations like using a pointer instead of popping from the preference list we would be optimal.



Above is the runtimes for the verification algorithm we implemented that checks for unstable pairs and other invalidities. It generally matches the trend seen in the previous graph, however it follows the n^2 trend closer.