Speed Comparison

N	Insertion	Knuth	Hibbard	Pratt
100	46	24	22	32
200	172	63	66	79
300	449	124	122	190
400	922	169	196	371
500	1591	270	301	550
600	2630	407	456	799
700	3939	576	609	1103
800	5676	755	792	1501
900	7793	964	1018	1915
1000	10531	1198	1300	2400

While all Shell Sort implementations performed better than Insertion Sort, the Knuth implementation was a little faster than the rest. This is likely due to the fact that there were larger gaps (and therefore fewer number of passes) in the Knuth gap sequence than the Hibbard and Pratt gap sequences.

Also, my Pratt implementation required a lot of overhead, i.e. completing multiple retrievals and comparisons on the gap sequence to create the ordered list without duplicates. This wouldn't affect the number of comparisons and assignments done in the sorting process, but this affects the overall speed.

Finally, all three Shell Sort implementations were faster than Insertion Sort, especially for larger lists, proving the validity of the implementations and algorithm.