CS 7345 Sorting Library Performance Testing:

(Average Time in seconds)	**Bubble**	**Merge**	**Insertion**
Reversed (10)	$2.22 * 10^{-5}$	$1.17 * 10^{-4}$	$2.08 * 10^{-5}$
Reversed (1,000)	0.033	0.00748	0.0108
Reversed (10,000)	3.140	0.0792	1.030
Reversed (100,000)	314.0	0.836	102.01

(Average Time in seconds)	**Bubble**	**Merge**	**Insertion**
100% Random (10)	$4.6 * 10^{-4}$	$1.1 * 10^{-4}$	$1.79 * 10^{-5}$
100% Random (1,000)	0.018	0.00729	0.00242
100% Random (10,000)	1.97	0.0811	0.191
100% Random (100,000)	209.2	0.8511	19.3

(Average Time in seconds)	**Bubble**	**Merge**	**Insertion**
80% Duplicate (10)	$1.66 * 10^{-5}$	$1.15 * 10^{-4}$	$2.81 * 10^{-5}$
80% Duplicate (1,000)	0.0204	0.00719	0.00274
80% Duplicate (10,000)	2.1203	0.0803	0.2411
80% Duplicate (100,000)	221.102	0.846	23.9

Total Average Runtime (All Algorithms on All Vector Types): *901.00 +/- 2.14 seconds*

Multi-Threaded:

(Average Time in seconds)	**Bubble**	**Merge**	**Insertion**
Reversed (10)	$5.96 * 10^{-4}$	$4.24 * 10^{-4}$	$4.08 * 10^{-4}$
Reversed (1,000)	0.0307	0.00819	0.0155
Reversed (10,000)	2.76	0.601	1.312
Reversed (100,000)	283.00	60.8	132.0

(Average Time in seconds)	**Bubble**	**Merge**	**Insertion**
100% Random (10)	0.0029	0.000536	$4.47 * 10^{-4}$
100% Random (1,000)	0.0197	0.00535	0.00712
100% Random (10,000)	1.72	0.377	0.521
100% Random (100,000)	178.01	40.6	53.14

(Average Time in seconds)	**Bubble**	**Merge**	**Insertion**
80% Duplicate (10)	$5.21 * 10^{-4}$	$4.24 * 10^{-4}$	$4.58 * 10^{-4}$
80% Duplicate (1,000)	0.0197	0.00556	0.00676
80% Duplicate (10,000)	1.73	0.418	0.534
80% Duplicate (100,000)	175.3	42.92	58.5

Total Average Runtime (All Algorithms on All Vector Types): *1070.1 +/- 71.9 seconds*

Multi-Threaded (Optimized)*[1]:

(Average Time in seconds)	Bubble	Merge	Insertion
Reversed (10)	$3.93 * 10^{-4}$	$3.29 * 10^{-4}$	$3.02 * 10^{-4}$
Reversed (1,000)	0.00234	$9.17 * 10^{-4}$	0.00173
Reversed (10,000)	0.163	0.0345	0.116
Reversed (100,000)	16.8	3.19	11.34

(Average Time in seconds)	Bubble	Merge	Insertion
100% Random (10)	0.0127	$3.17 * 10^{-4}$	$3.34 * 10^{-4}$
100% Random (1,000)	0.00341	$7.41 * 10^{-4}$	$9.63 * 10^{-4}$
100% Random (10,000)	0.159	0.0308	0.438
100% Random (100,000)	14.6	3.21	4.64

(Average Time in seconds)	Bubble	Merge	Insertion
80% Duplicate (10)	$3.64 * 10^{-4}$	$3.06 * 10^{-4}$	$3.71 * 10^{-4}$
80% Duplicate (1,000)	0.00189	$7.50 * 10^{-4}$	$9.54 * 10^{-4}$
80% Duplicate (10,000)	0.138	0.0313	0.0465
80% Duplicate (100,000)	13.82	3.01	4.62

Total Average Runtime (All Algorithms on All Vector Types): *76.2 +/- 4.13 seconds*

Client/Network:

(Average Time in seconds)	Bubble	Merge	Insertion
Reversed (10)	0.0381	0.0412	0.0379
Reversed (1,000)	0.0907	0.0672	0.0846
Reversed (10,000)	0.769	0.502	0.497
Reversed (100,000)	49.3	5.81	10.4

(Average Time in seconds)	Bubble	Merge	Insertion
100% Random (10)	0.0212	0.0398	0.0335
100% Random (1,000)	0.0876	0.802	0.0882
100% Random (10,000)	0.5440	0.444	0.355
100% Random (100,000)	35.8	6.12	10.4

(Average Time in seconds)	Bubble	Merge	Insertion
80% Duplicate (10)	0.0328	0.0421	0.0344

[1] Optimization with regards to Emscripten means employing the minification process with the compiler flag "-O2" when creating the document. More information about how Emscripten optimizes code can be found here: https://emscripten.org/docs/optimizing/Optimizing-Code.html

80% Duplicate (1,000)	0.0997	0.0783	0.0902
80% Duplicate (10,000)	0.753	0.510	0.52
80% Duplicate (100,000)	41.5	6.53	14.7

Total Average Runtime (All Algorithms on All Vector Types): *183.63 +/- 4.38 seconds*