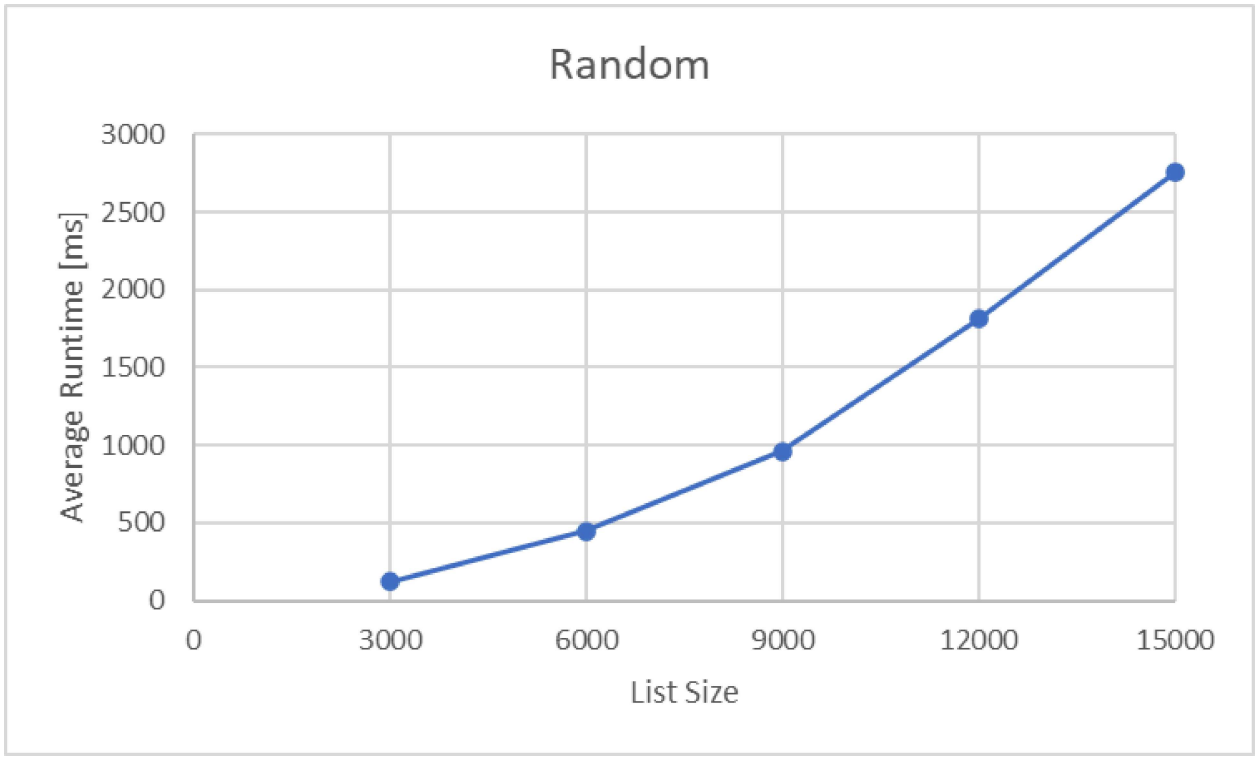
MP3 Test Plan

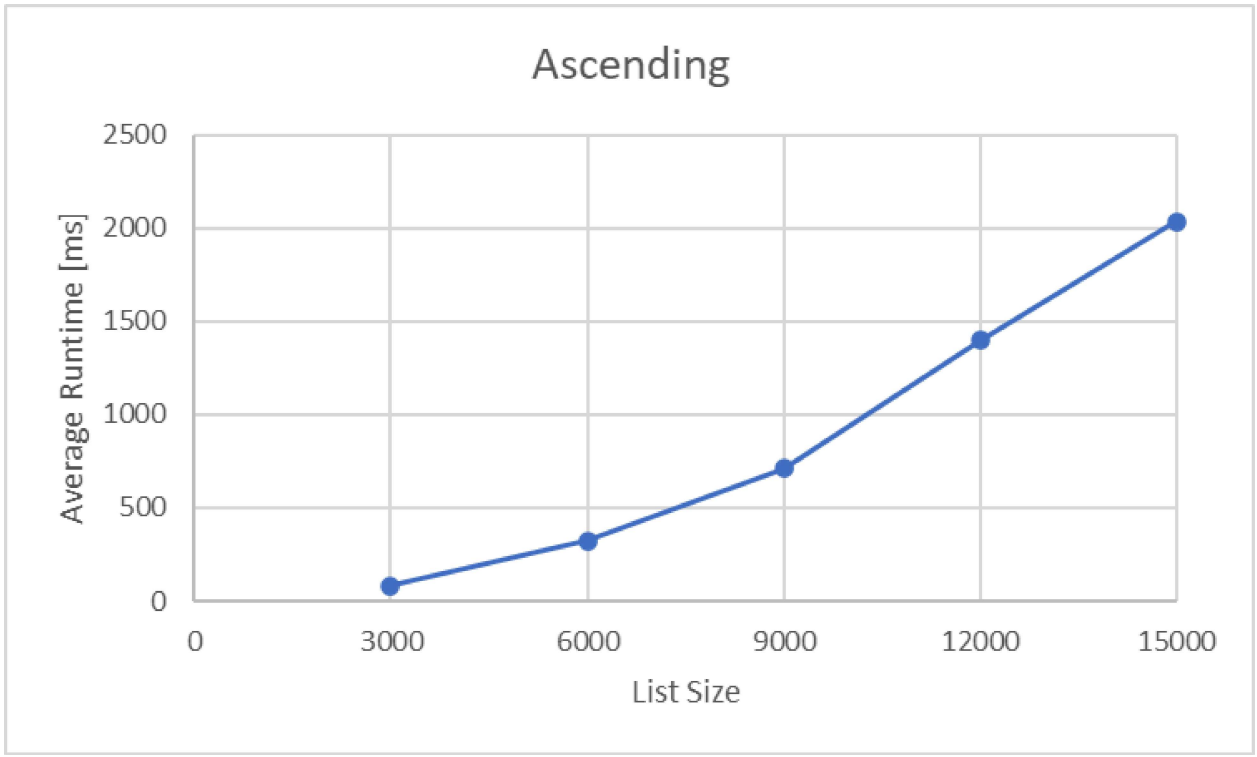
Test Type Performances

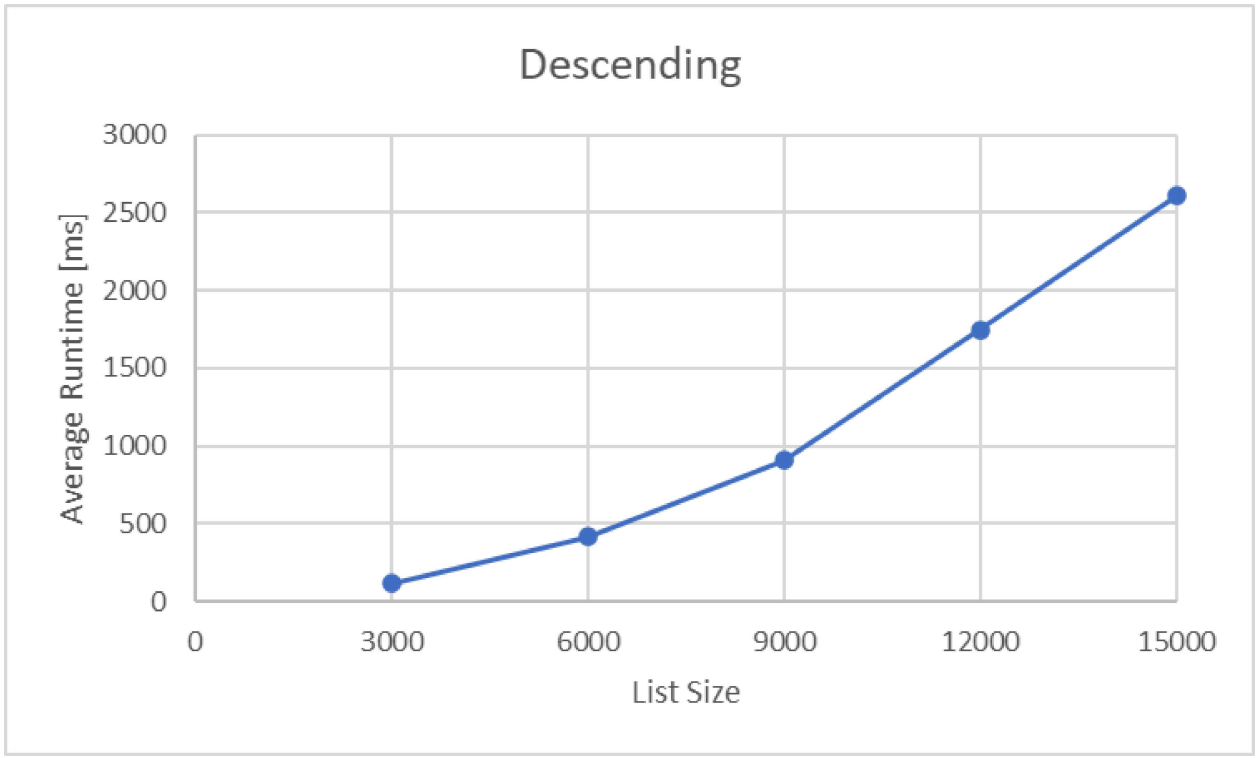
This section outlines the test cases designated for each of the 5 sorting functions implemented in MP3. Each function undergoes testing with randomly ordered input, ascending input, and descending input. Across each input scenario, 5 different input sizes are tested, carefully selected to provide a diverse range of runtime values, ranging from less than a hundred milliseconds to over 1 second. Due to runtime variability, each case is tested three times, and the average runtime of these iterations is presented.

1. Bubble Sort

List Type and Size	1st Runtime	2nd Runtime	3rd Runtime	Average Runtime
Random 3000	116.48	123.08	115.55	118.37
Random 6000	428.64	479.00	429.02	445.55
Random 9000	963.96	956.91	957.28	959.38
Random 12000	1819.42	1820.68	1802.20	1814.10
Random 15000	2720.81	2770.35	2776.62	2755.92
Ascending 3000	80.03	81.03	82.74	81.27
Ascending 6000	321.29	325.03	321.15	322.49
Ascending 9000	707.96	713.85	709.45	710.42
Ascending 12000	1439.44	1404.05	1356.30	1399.93
Ascending 15000	2027.35	2036.35	2043.60	2035.77
Descending 3000	113.00	113.38	114.86	113.75
Descending 6000	420.91	413.11	407.80	413.94
Descending 9000	900.89	909.06	903.88	904.61
Descending 12000	1697.42	1818.87	1718.07	1744.79
Descending 15000	2651.01	2587.02	2595.03	2611.02

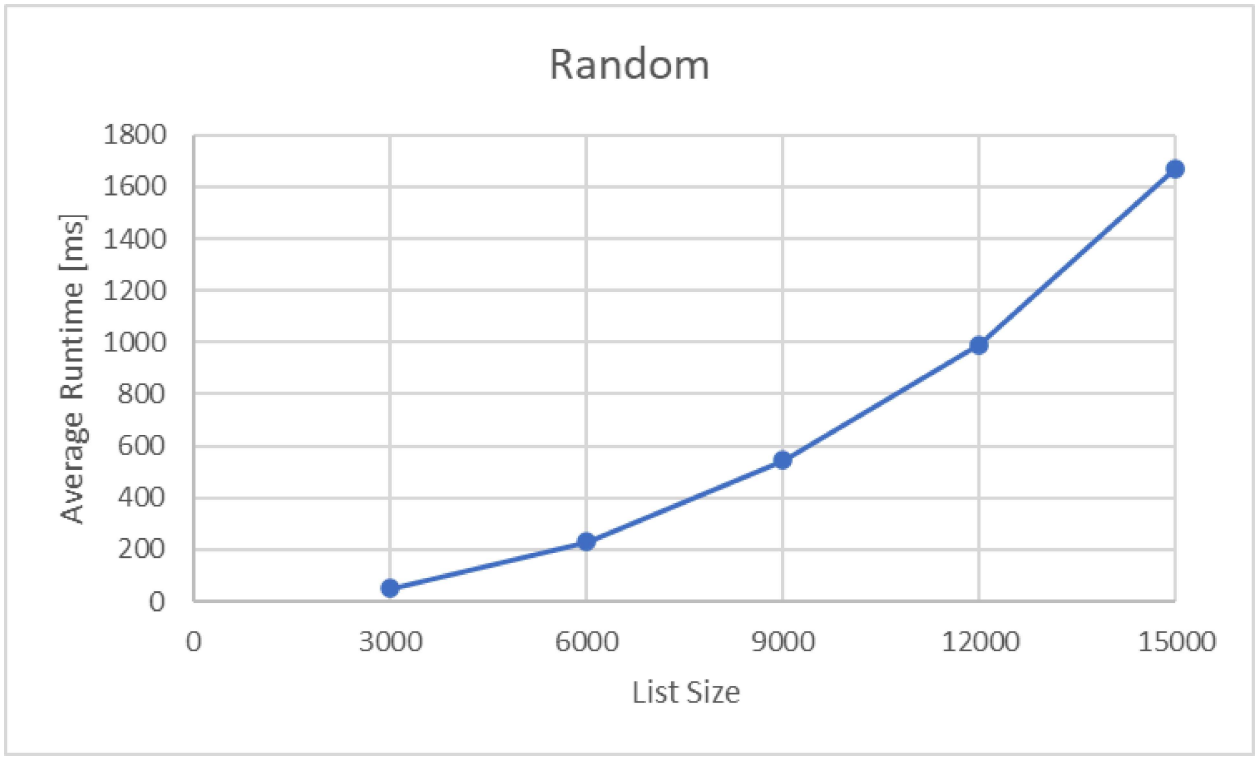

Random
Average Runtime [ms]
3000
2500
2000
1500
1000
500
0
0
3000
6000
9000
12000
15000
List Size

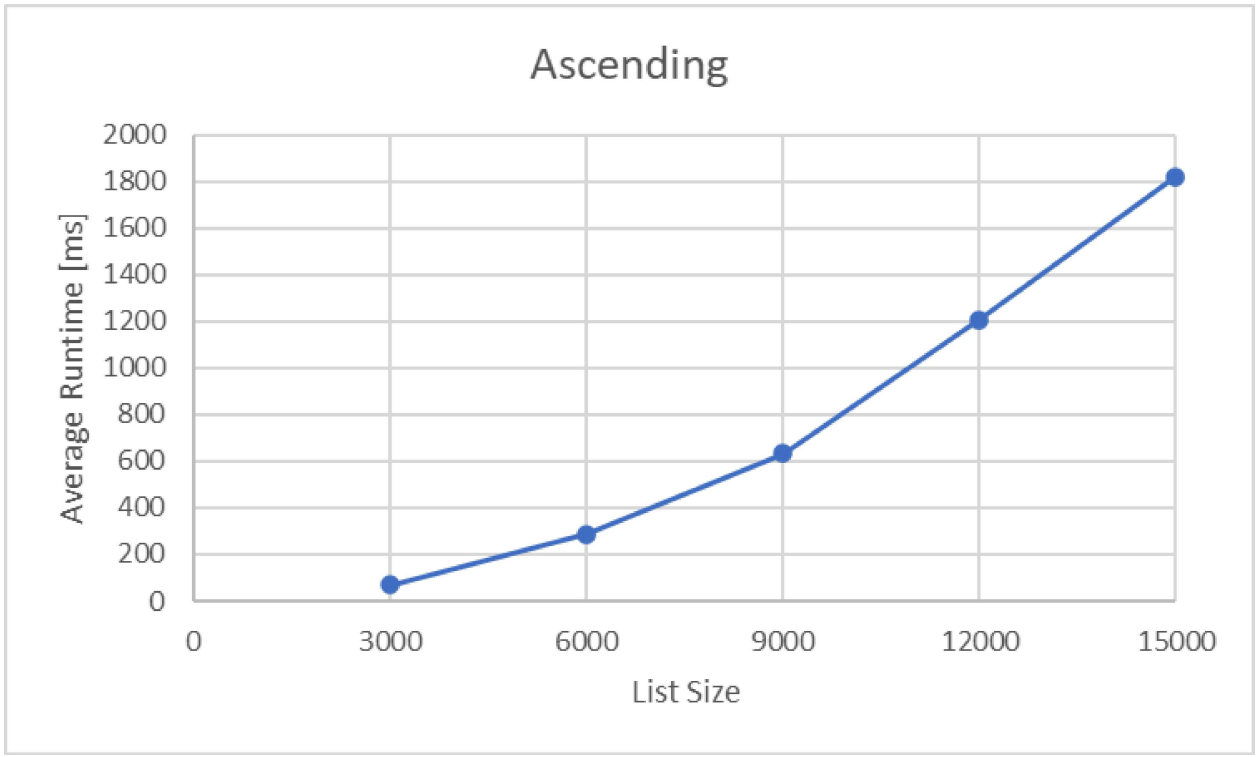

Ascending
Average Runtime [ms]
2500
2000
1500
1000
500
0
0
3000
6000
9000
12000
15000
List Size


Descending
Average Runtime [ms]
3000
2500
2000
1500
1000
500
0
0
3000
6000
9000
12000
15000
List Size

2. Insertion Sort

List Type and Size	1st Runtime	2nd Runtime	3rd Runtime	Average Runtime
Random 3000	48.74	47.93	50.32	49.00
Random 6000	235.26	227.58	229.12	230.65
Random 9000	543.90	531.80	554.33	543.34
Random 12000	1000.98	993.84	979.43	991.42
Random 15000	1678.48	1678.97	1655.57	1671.01
Ascending 3000	70.66	69.51	70.11	70.09
Ascending 6000	288.27	283.71	291.83	287.94
Ascending 9000	622.87	645.42	627.52	631.94
Ascending 12000	1205.02	1202.88	1213.76	1207.22
Ascending 15000	1832.75	1824.92	1806.51	1821.39
Descending 3000	0.32	0.27	0.30	0.30
Descending 6000	0.72	0.55	0.67	0.65
Descending 9000	0.99	1.04	0.88	0.97
Descending 12000	1.33	1.24	1.09	1.22
Descending 15000	1.69	1.63	1.46	1.59

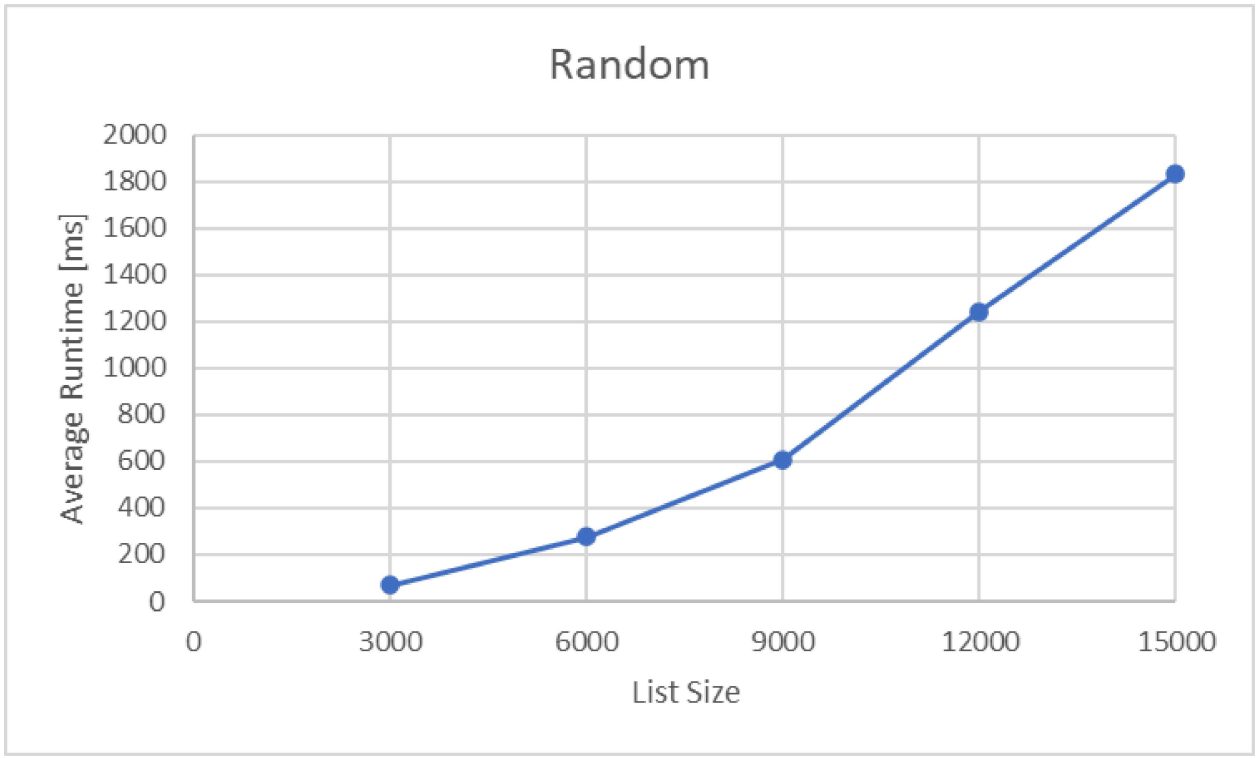

Random
Average Runtime [ms]
1800
1600
1400
1200
1000
800
600
400
200
0
0
3000
6000
9000
12000
15000
List Size

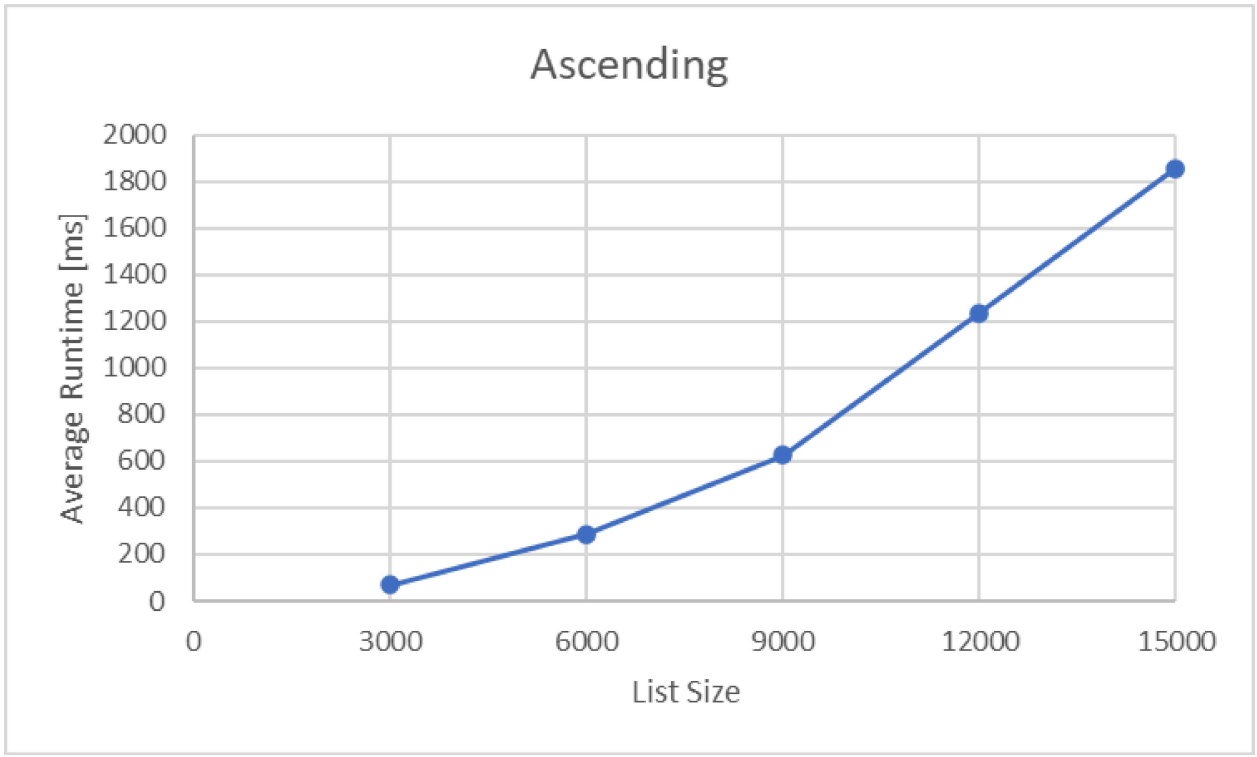

Ascending
Average Runtime [ms]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
3000
6000
9000
12000
15000
List Size

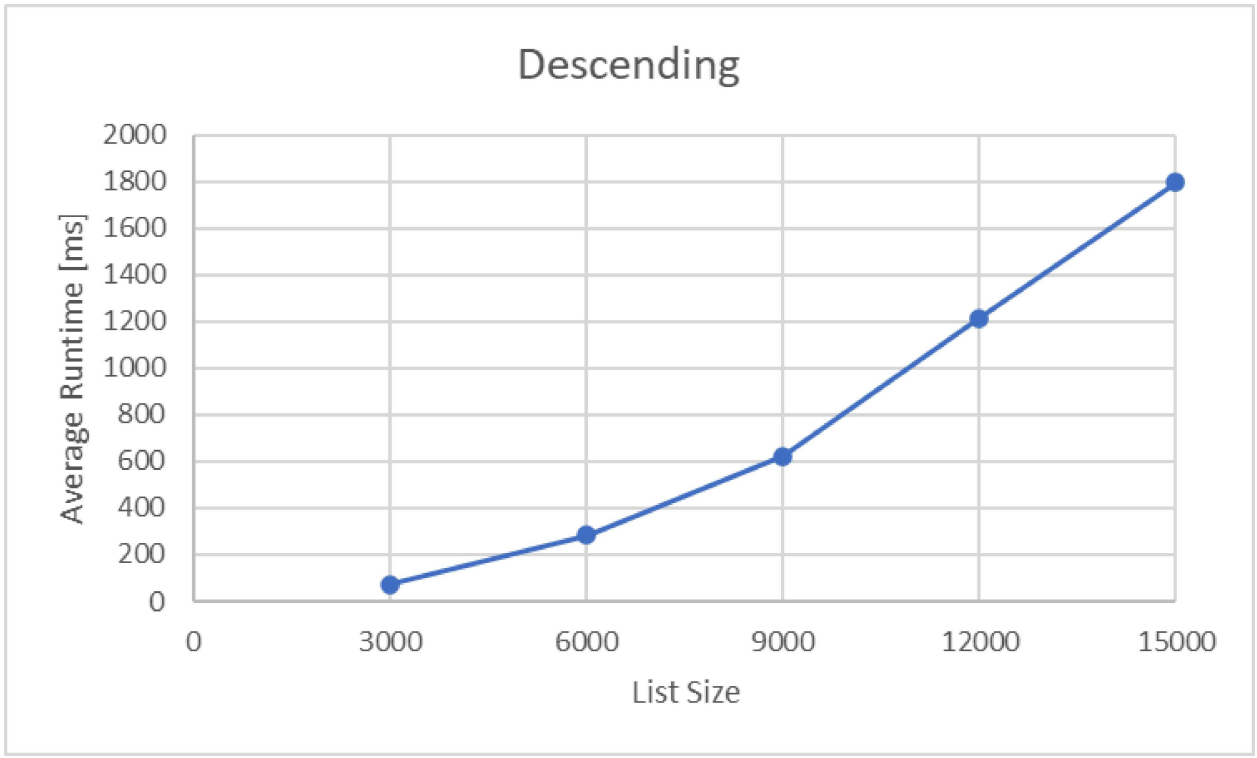

Descending
Average Runtime [ms]
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2000
4000
6000
8000
10000
12000
14000
16000
List Size

3. Recursive Selection Sort

List Type and Size	1st Runtime	2nd Runtime	3rd Runtime	Average Runtime
Random 3000	70.18	68.66	69.32	69.39
Random 6000	275.69	274.90	278.20	276.26
Random 9000	600.08	607.55	611.78	606.47
Random 12000	1164.82	1253.27	1310.0	1242.69
Random 15000	1771.32	1942.08	1778.60	1830.67
Ascending 3000	71.66	69.14	70.17	70.32
Ascending 6000	282.44	289.15	283.90	285.16
Ascending 9000	619.70	626.50	626.76	624.32
Ascending 12000	1191.18	1213.71	1301.27	1235.39
Ascending 15000	1897.88	1812.38	1860.96	1857.07
Descending 3000	73.71	70.79	71.83	72.11
Descending 6000	283.73	282.77	286.53	284.34
Descending 9000	627.03	620.14	624.94	624.04
Descending 12000	1196.48	1233.62	1210.43	1213.51
Descending 15000	1781.01	1802.50	1803.96	1795.82

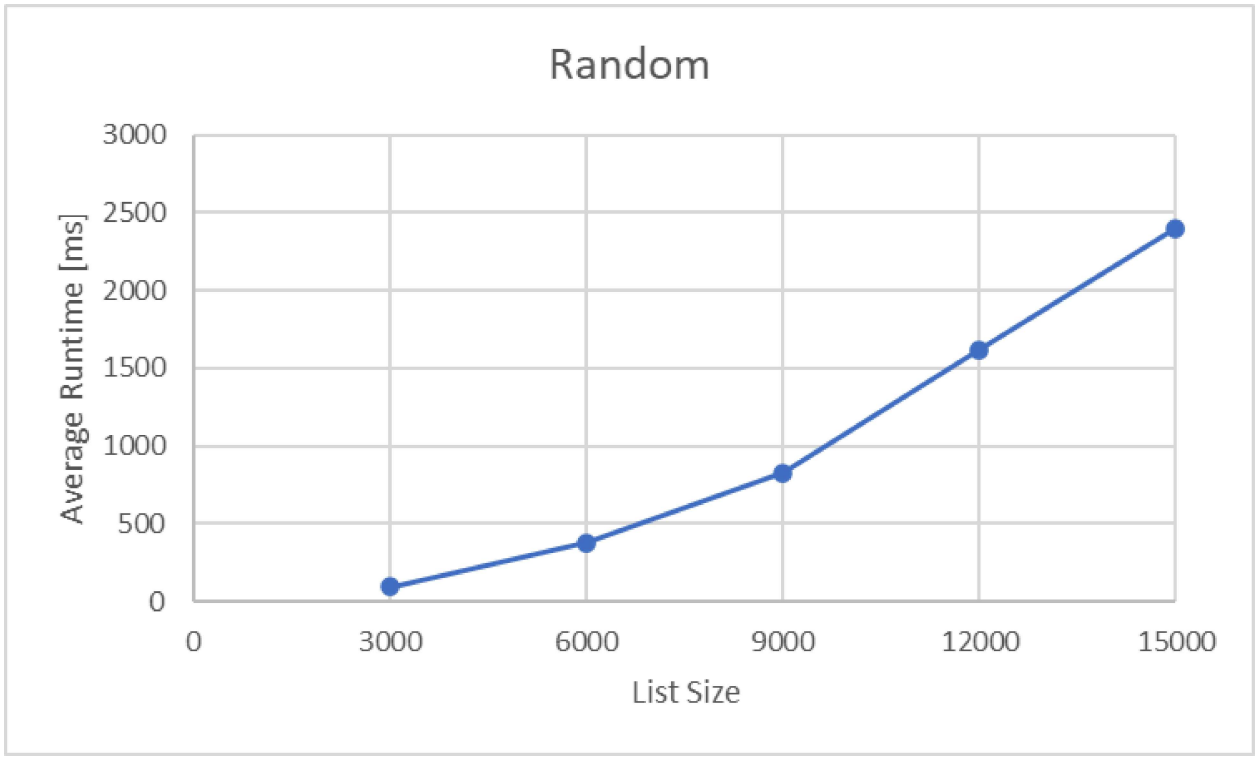

Random
Average Runtime [ms]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
3000
6000
9000
12000
15000
List Size

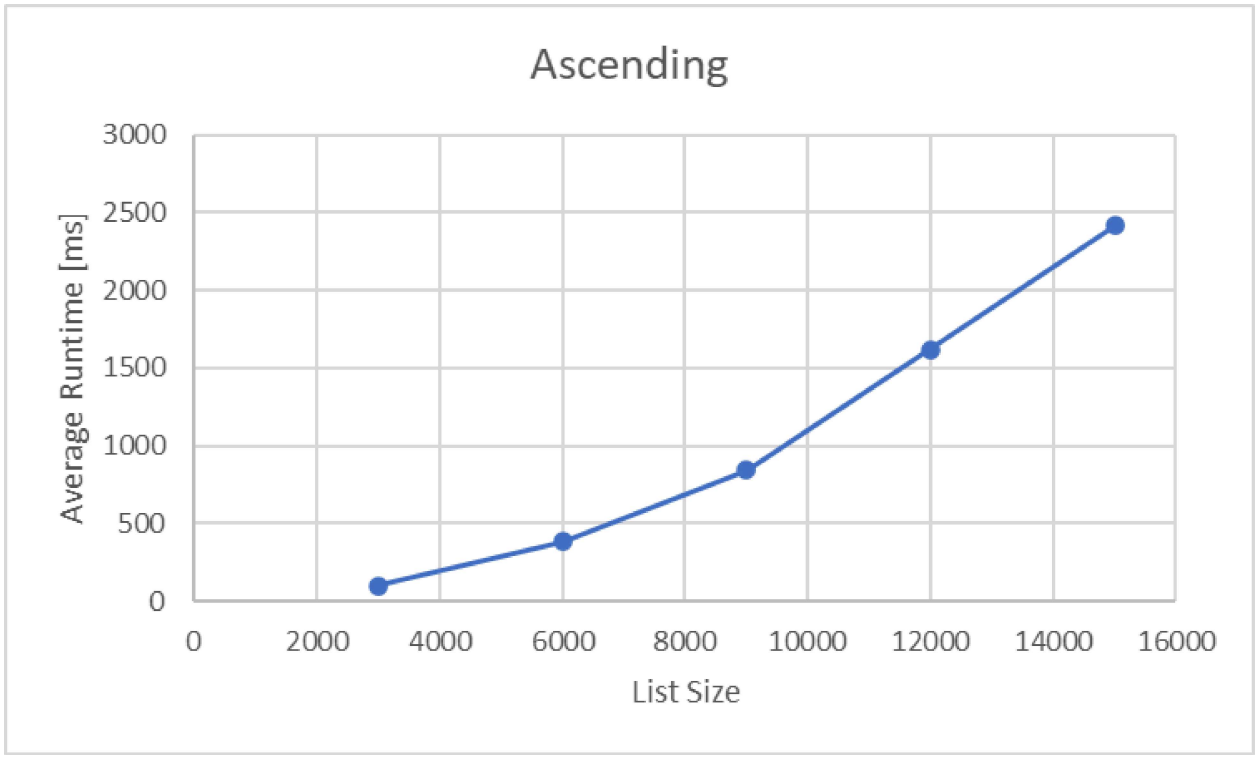

Ascending
Average Runtime [ms]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
3000
6000
9000
12000
15000
List Size

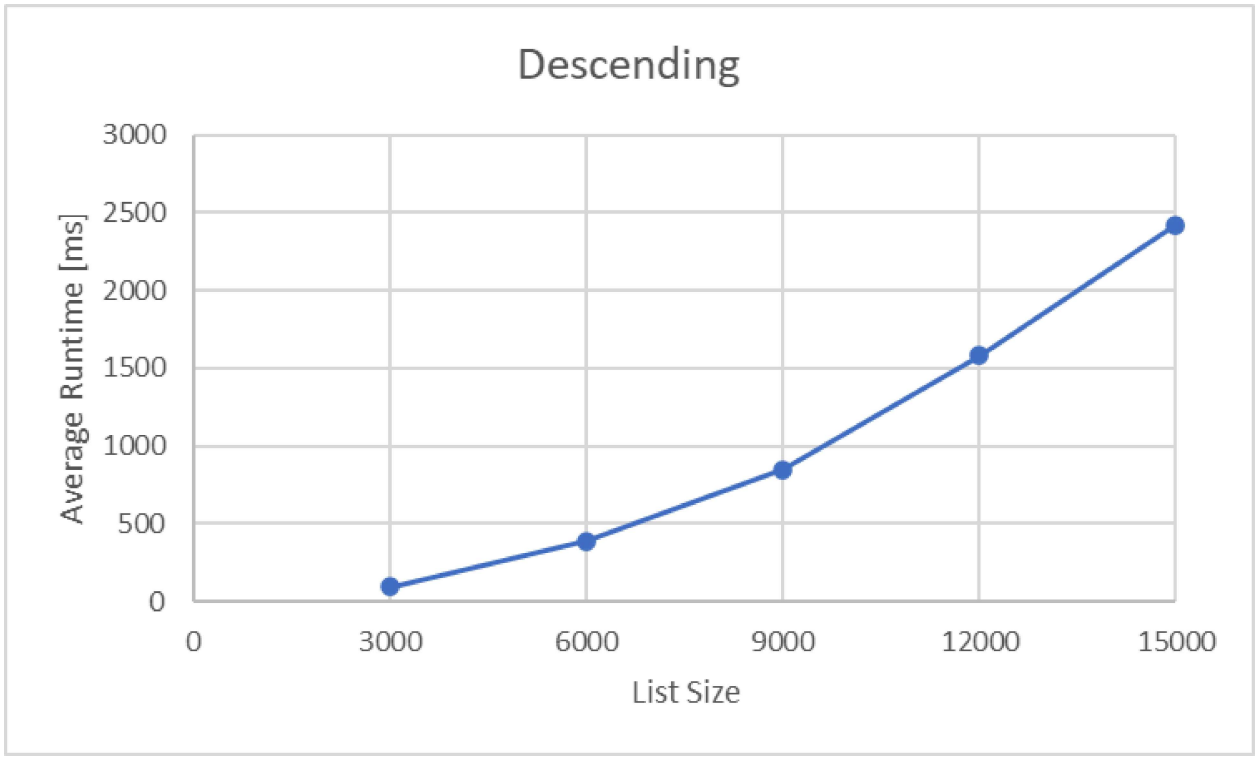

Descending
Average Runtime [ms]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
3000
6000
9000
12000
15000
List Size

4. Iterative Selection Sort

List Type and Size	1st Runtime	2nd Runtime	3rd Runtime	Average Runtime
Random 3000	95.38	93.65	93.29	94.11
Random 6000	379.74	373.55	378.14	377.14
Random 9000	831.86	820.23	820.23	824.11
Random 12000	1576.31	1719.56	1566.65	1620.84
Random 15000	2401.68	2394.73	2399.79	2398.73
Ascending 3000	95.13	98.05	97.63	96.94
Ascending 6000	380.78	381.24	378.84	380.29
Ascending 9000	840.44	842.09	840.16	840.89
Ascending 12000	1692.96	1588.77	1580.30	1620.67
Ascending 15000	2401.79	2428.87	2413.38	2414.68
Descending 3000	95.22	95.59	96.41	95.74
Descending 6000	382.91	379.04	394.23	385.39
Descending 9000	837.24	865.49	833.70	845.48
Descending 12000	1581.31	1583.44	1581.95	1582.23
Descending 15000	2456.07	2401.92	2409.70	2422.56

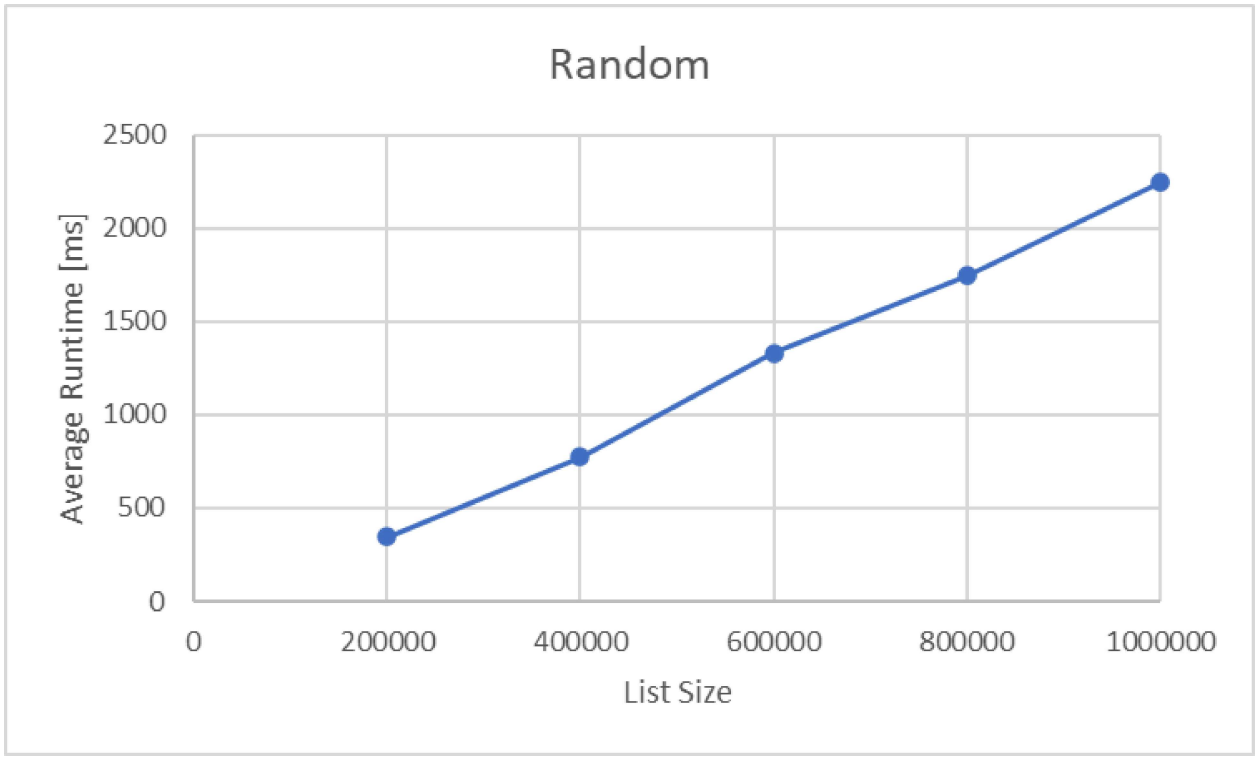

Random
Average Runtime [ms]
3000
2500
2000
1500
1000
500
0
0
3000
6000
9000
12000
15000
List Size

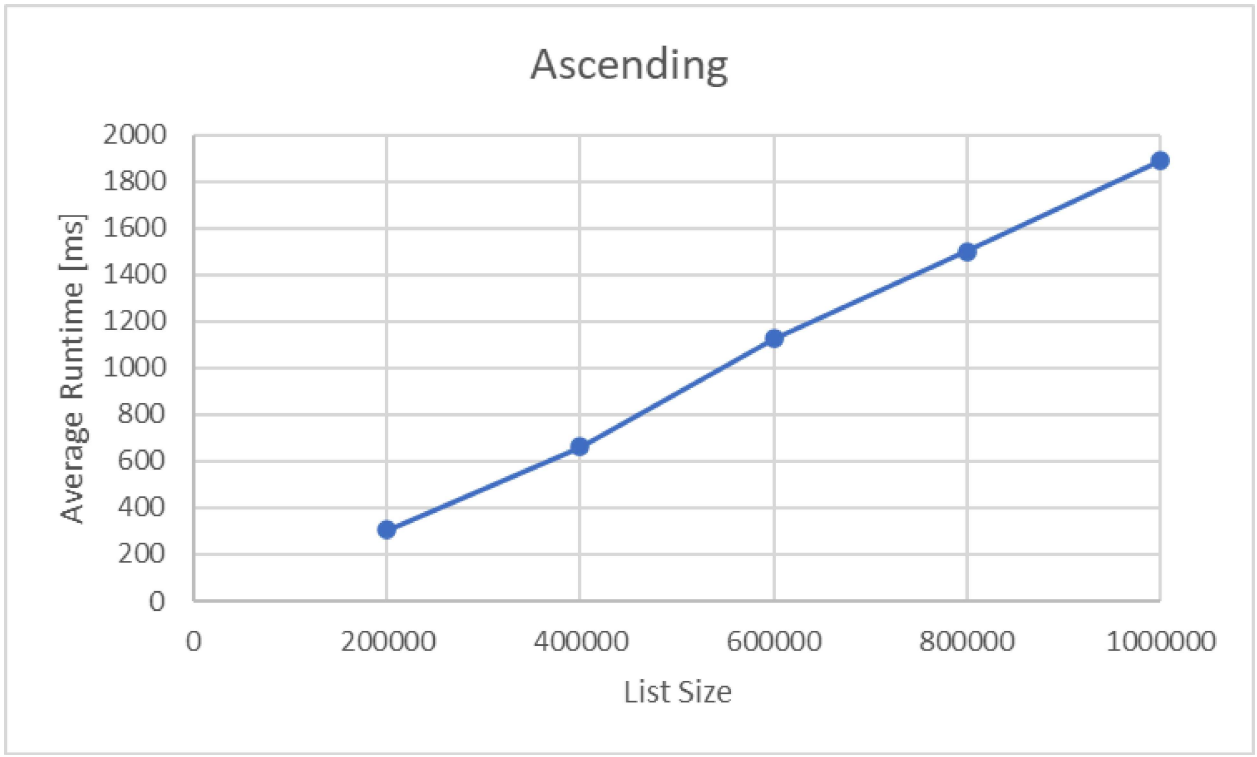

Ascending
Average Runtime [ms]
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
14000
16000
List Size

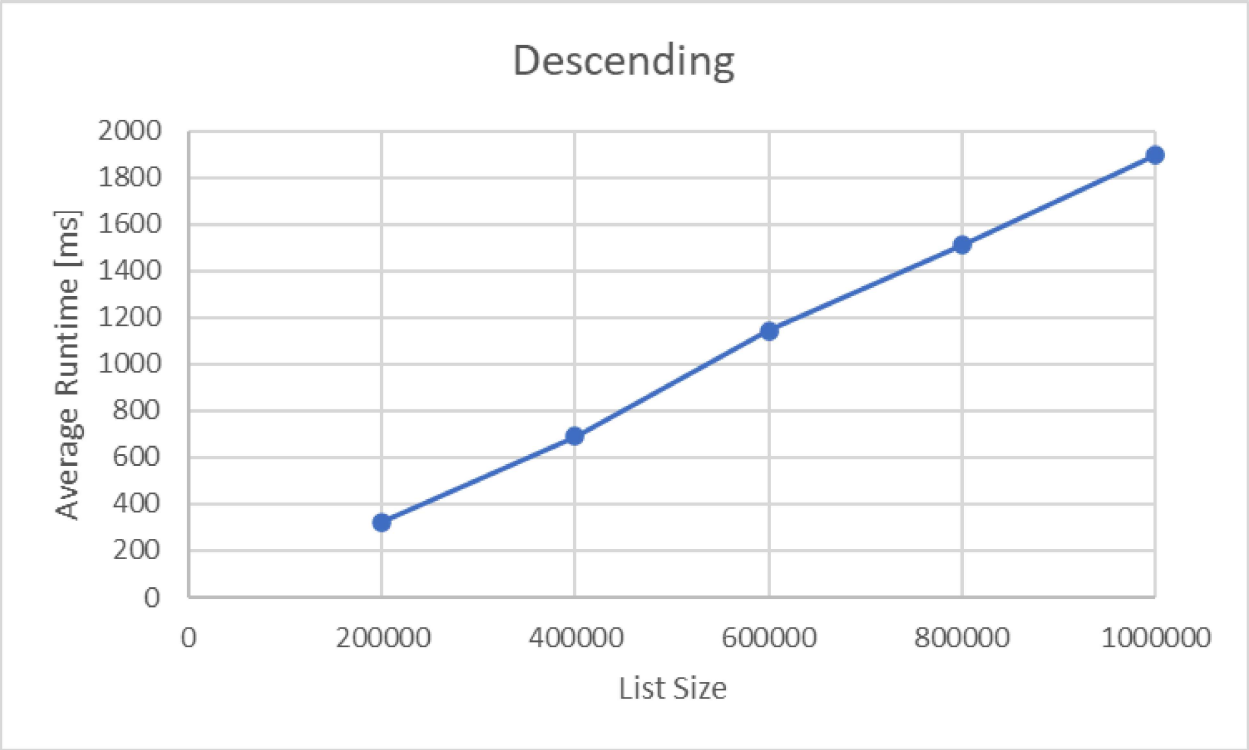

Descending
Average Runtime [ms]
3000
2500
2000
1500
1000
500
0
0
3000
6000
9000
12000
15000
List Size

5. Merge Sort

List Type and Size	1st Runtime	2nd Runtime	3rd Runtime	Average Runtime
Random 200000	344.47	346.67	345.31	345.48
Random 400000	765.37	780.83	772.30	772.83
Random 600000	1318.98	1322.70	1348.39	1330.02
Random 800000	1772.31	1662.75	1808.92	1747.99
Random 1000000	2254.37	2245.91	2241.72	2247.33
Ascending 200000	309.03	303.07	303.47	305.19
Ascending 400000	653.58	672.83	653.66	660.02
Ascending 600000	1123.30	1126.36	1123.80	1124.49
Ascending 800000	1503.30	1496.84	1500.38	1500.17
Ascending 1000000	1861.58	1918.21	1886.63	1888.81
Descending 200000	311.71	306.51	350.219	322.81
Descending 400000	668.29	689.01	712.41	689.90
Descending 600000	1128.38	1146.92	1157.90	1144.4
Descending 800000	1502.87	1509.22	1523.57	1511.89
Descending 1000000	1899.14	1878.20	1909.10	1895.48


Random
Average Runtime [ms]
2500
2000
1500
1000
500
0
0
200000
400000
600000
800000
1000000
List Size


Ascending
Average Runtime [ms]
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
200000
400000
600000
800000
1000000
List Size



Analysis of Results

For initially randomized lists, the first 4 functions show similar runtimes, around 100 milliseconds for 3000 elements and 1.5-2 seconds for 15000 elements. Both recursive and iterative selection sort perform comparably. Despite differing implementations, their time complexities remain $O(n^2)$, resulting in similar runtimes. In contrast, merge sort exhibits a substantial improvement, being much faster due to its $O(n\log_n)$ complexity. This allows it to handle much larger lists within the same timeframe.

Bubble sort shows slight performance gains when the input list is already sorted. In insertion sort, a significant runtime decrease occurs when sorting a descending list into ascending order due to reduced comparisons. No notable runtime differences are observed in selection sort or merge sort based on the input list's initial sorting.