10mb 100mb

	Unix Test1.txt			Unix Test2.txt		
	Real	User	Sys	Real	User	Sys
Trial 1	6.114	2.016	1.204	65.518	26.291	11.961
Trial 2	5.400	2.013	1.204	60.152	26.121	11.991
Trial 3	5.280	1.997	1.189	60.028	26.176	11.981
Average	5.598	2.009	1.199	61.899	26.196	11.978

	Bucket Sort Test1.txt			Bucket Sort Test2.txt		
	Real	User	Sys	Real	User	Sys
Trial 1	13.350	10.597	2.953	136.744	98.474	28.460
Trial 2	17.074	11.438	3.075	122.661	97.517	28.423
Trial 3	18.932	11.951	3.046	126.953	98.278	28.860
Average	16.452	11.329	3.025	128.786	98.090	28.581

	Counting Sort Test1.txt			Counting Sort Test2.txt		
	Real	User	Sys	Real	User	Sys
Trial 1	14.787	9.987	2.845	100.237	117.954	29.190
Trial 2	16.225	10.597	2.929	153.423	117.423	29.252
Trial 3	18.351	10.761	2.888	172.872	121.939	29.060
Average	16.454	10.448	2.887	142.177	119.105	29.167

As expected the default unix sort beat both of the java implementations of radix/bucket sort and radix/counting sort by a wide margin. Bucket sort and counting sort are very close to eachother. so much so that they are only 2/100ths of a second off from eachother in the 10mb file. When sorting the 100mb file bucket sort finshed 14 seconds before counting sort. Neither of these are very surprisng considering how closely related the two sorts are. The difference is most likely explained by the differnece in overhead for each of the functions.