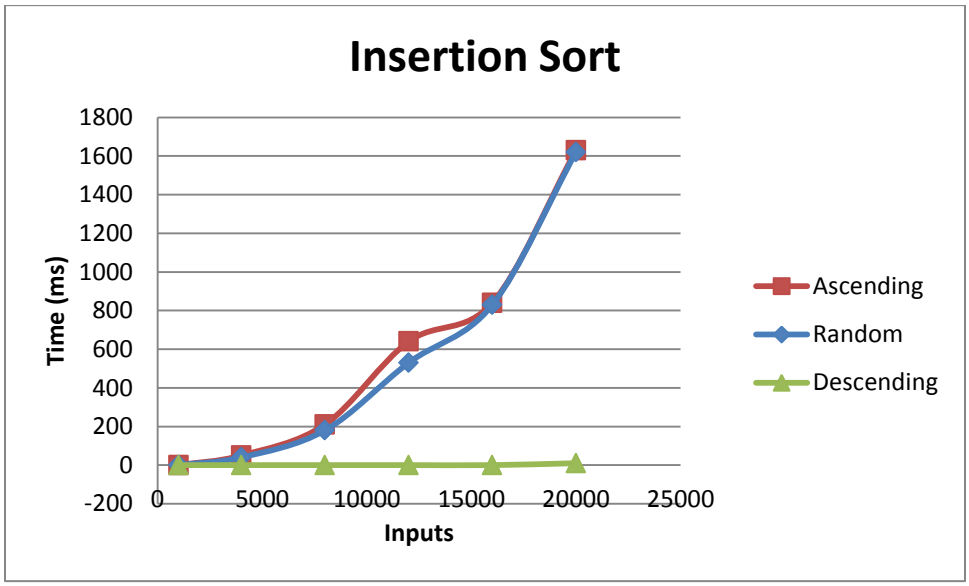
Denzel Murdaugh

Test Log

Known bugs: After a lot of calls there is a function call error in merge;
Program received signal SIGSEGV, Segmentation fault.
0x0000000000401f50 in trte_compare (
    record_a=<error reading variable: Cannot access memory at address 0x7fffff7feff8>,
    record_b=<error reading variable: Cannot access memory at address 0x7fffff7feff0>) at trte_support.c:56
56   {
From what I can tell it works fine, just after too many calls there is some weird corruption of the inputs. If I manually input the compare function into the merge() function then I can get a bit farther into the list but still run into the same problem.

Insertion Sort

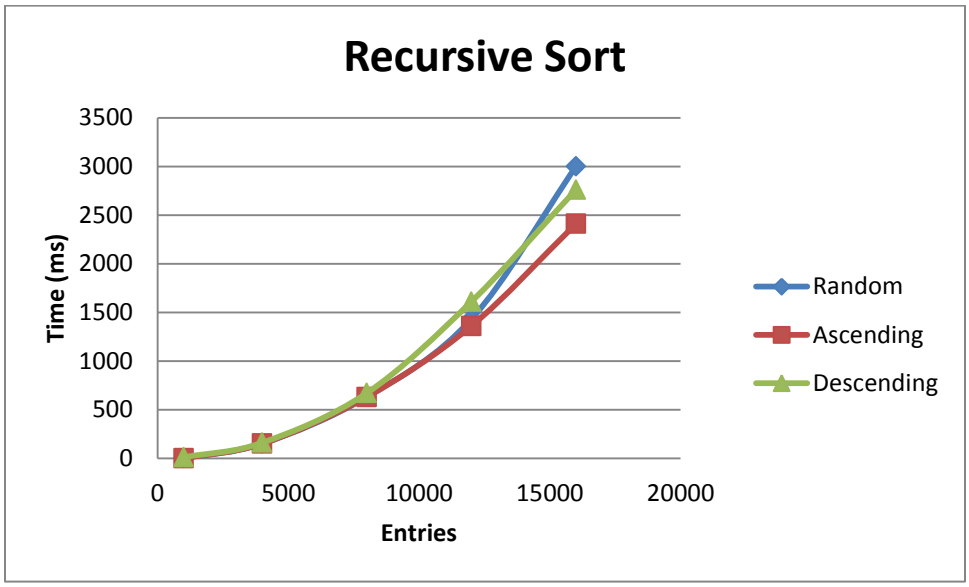
Entries	Random	Ascending	Descending
1000	0	0	0
4000	40	50	0
8000	180	210	0
12000	530	640	0
16000	830	840	0
20000	1620	1630	10



Denzel Murdaugh

Recursive Sort

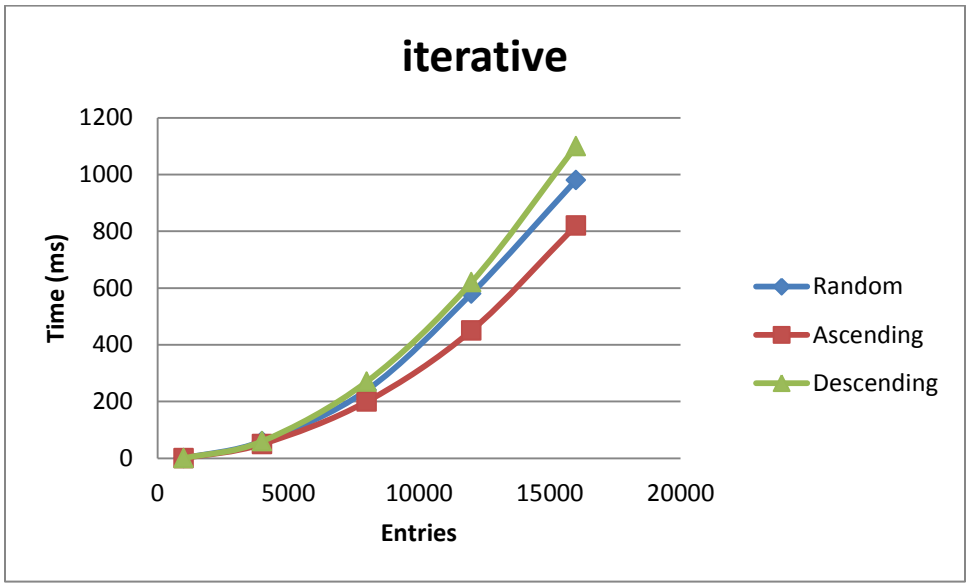
Entries	Random	Ascending	Descending
1000	0	0	10
4000	150	150	160
8000	630	630	670
12000	1430	1360	1610
16000	3000	2410	2760



Denzel Murdaugh

Iterative Selection

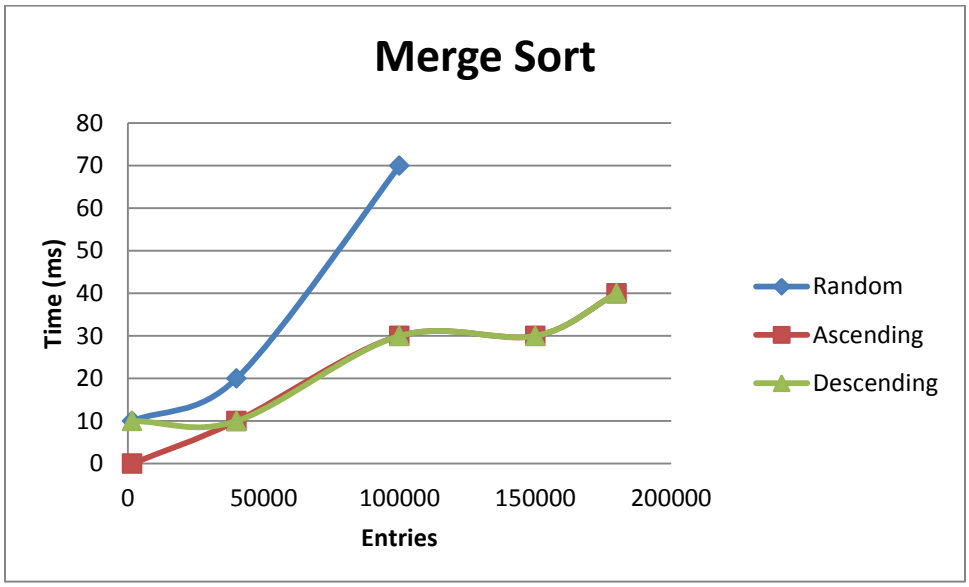
Entries	Random	Ascending	Descending
1000	0	0	0
4000	60	50	60
8000	240	200	270
12000	580	450	620
16000	980	820	1100



Denzel Murdaugh

Merge

Entries	Random	Ascending	Descending
1600	10	0	10
40000	20	10	10
100000	70	30	30
150000		30	30
180000		40	40



a) There seems to be no real difference when sorting the random list when it comes to iterative and recursive sort. There's really no big difference between the iterative and recursive among any of my lists. MergeSort is so much faster than all the others because it splits the list into smaller pieces that it searches through instead of having to go all the way back to the front of 10,000 input list and search through it repeatedly.

b) If the list is already sorted the insertion sort flies through the list much faster than the other sorts besides mergeSort. This is because since the list is already sorted the insertion sort is essentially just replicating the list.