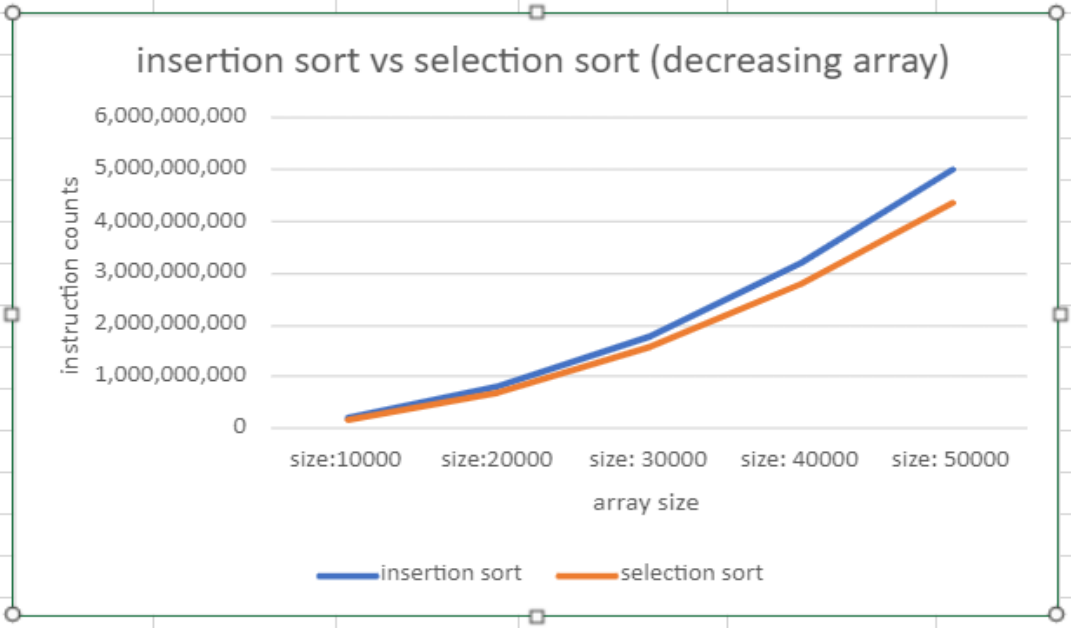
	A	B	C	D	E	F
1		size:10000	size:20000	size: 30000	size: 40000	size: 50000
2	insertion sort	200,039,996	800,079,996	1,800,119,996	3,200,159,996	5,000,199,996
3	selection sort	175,054,995	700,109,995	1,575,164,995	2,800,219,995	4,375,274,995



Contrary to the other 2 array orientations, the initially decreasing orientation sees the selection sort algorithm outperform its counterpart for all array sizes. For both algorithms, an initially decreasing array is the worst possible scenario. However, due to the fact that the selection sort algorithm must loop through and check every element in the unsorted portion of the array regardless of the array's initial orientation, the worst possible case has a much smaller effect on the selection sort algorithm than the insertion sort algorithm. The insertion sort algorithm's main advantage over the selection sort algorithm is that the insertion sort only requires 1 comparison per iteration of the inner loop, comparing a chosen value to the value in the cell immediately left to it. The initially decreasing set of data takes away this advantage, as the algorithm must now constantly compare and shift values to the right throughout the entire array. When it comes to the worst possible case, the selection sort algorithm executes significantly quicker than the insertion sort algorithm.