Assignment 3

---

1. Conclusion:

For a ordered array, Insertion sort has a much better performance than selection sort.

In random array sort, 2 methods has similar performance, but when the elements in array is large, Selection is better.

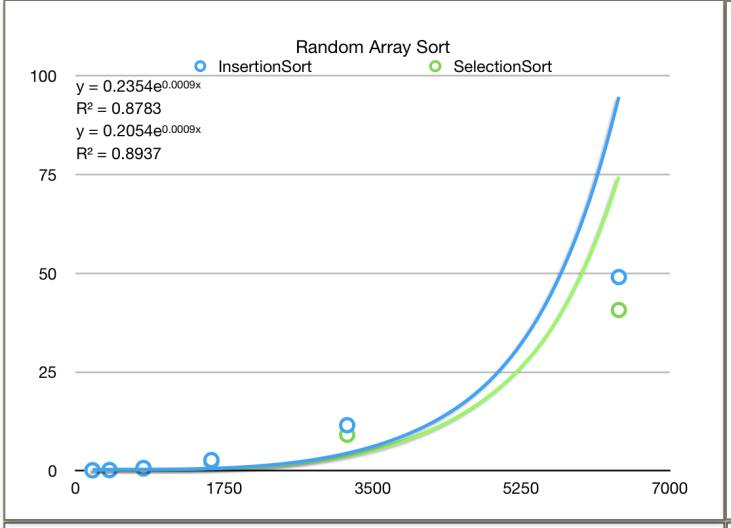
For a partially-ordered arrays, insertion sorting have a better performance when the quantity of sorting elements in array is bigger. They has little difference when the array is small.

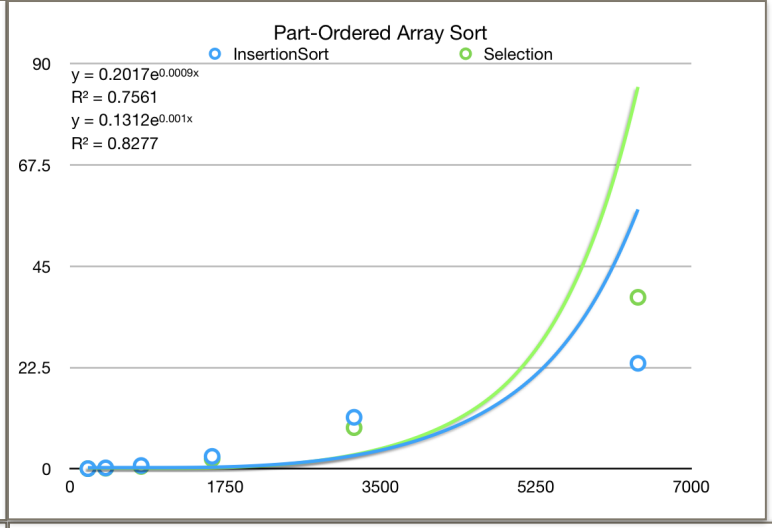
For a reversed-array, Selection performs better. The bigger array , the bigger difference.

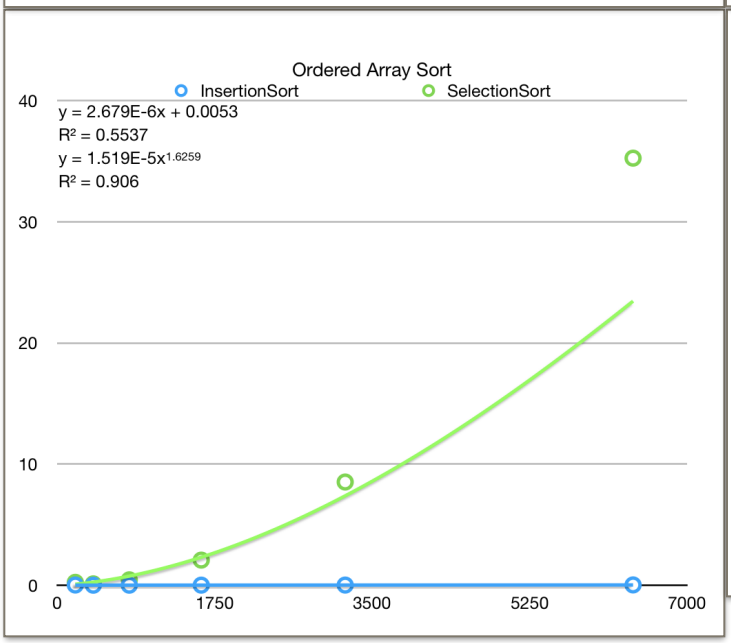
2. Data chart and trend line.

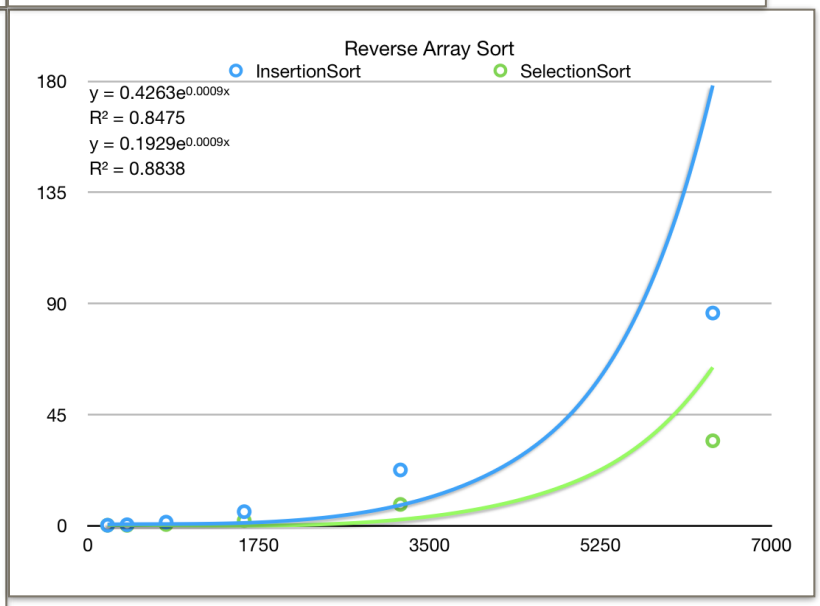
Different Array Sort Results by running 100 times

	Ordered		Reverse		Random		Part_Random	
	InsertionSort	SelectionSort	InsertionSort	SelectionSort	InsertionSort	SelectionSort	InsertionSort	SelectionSort
200	0.016588	0.238471	0.157611	0.154397	0.112492	0.150499	0.043802	0.034729
400	0.002095	0.114523	0.355293	0.112225	0.191227	0.133113	0.196933	0.12521
800	0.003719	0.445262	1.414505	0.445118	0.695166	0.489819	0.723214	0.519604
1600	0.006476	2.071829	5.654063	2.085902	2.699746	2.259659	2.753628	2.215552
3200	0.01257	8.517767	22.549918	8.589221	11.535054	9.09337	11.437007	9.136123
6400	0.02435	35.248007	86.107771	34.367914	49.035045	40.692419	23.430427	38.086265

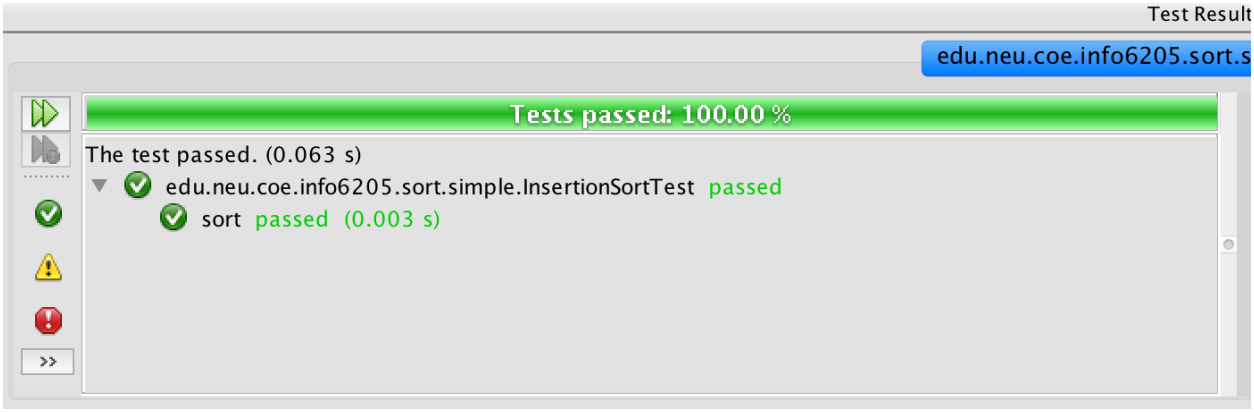




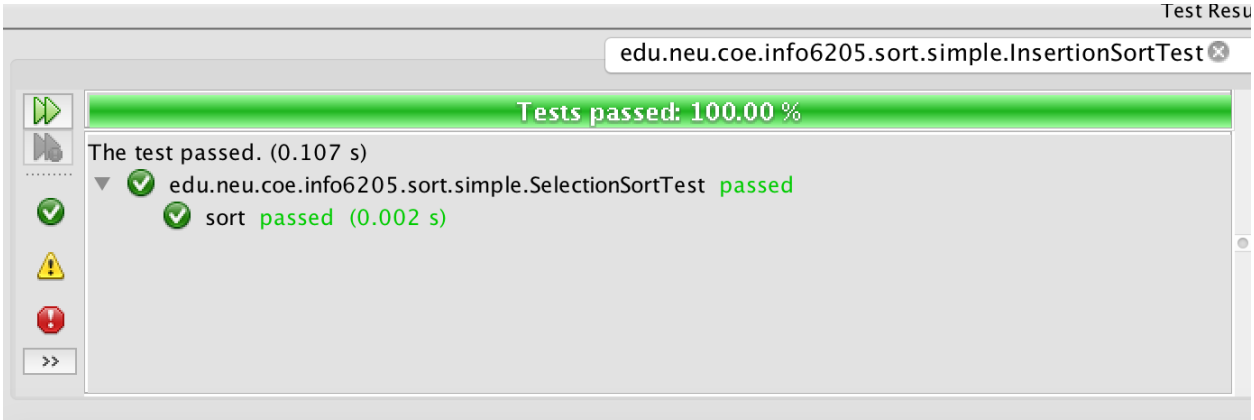




Trendline



Test results:


Test Resu
edu.neu.coe.info6205.sort.simple.InsertionSortTest
Tests passed: 100.00 %
The test passed. (0.107 s)
edu.neu.coe.info6205.sort.simple.SelectionSortTest passed
sort passed (0.002 s)
>>