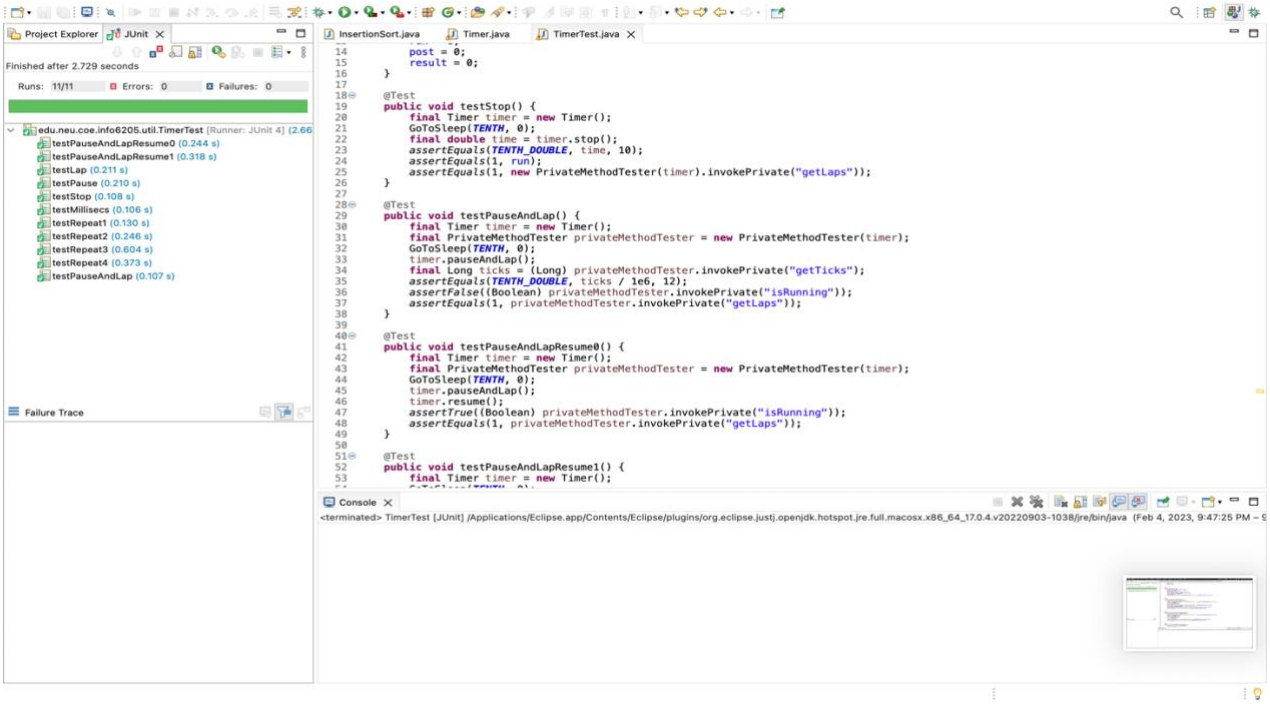
Program Structures and Algorithms Sec -8

Name: Uday Kiran Reddy Mulpuri **NUID:** 002781063

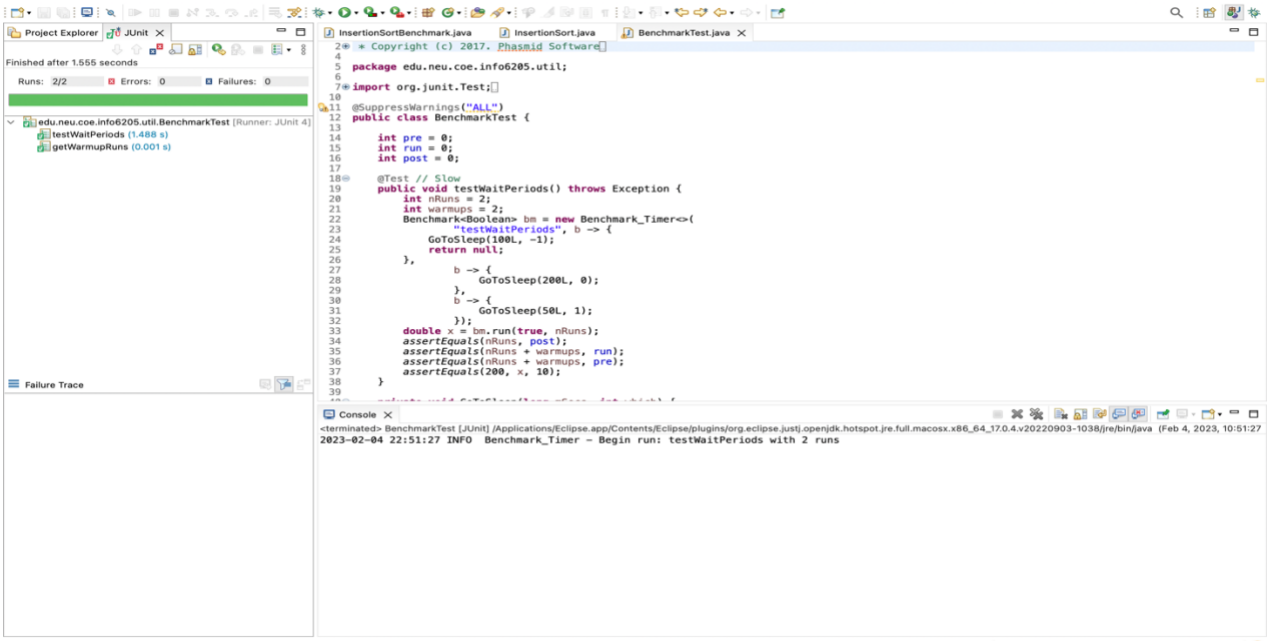
Assignment-3

Task 1: (Part 1) You are to implement three (3) methods (*repeat*, *getClock*, and *toMillisecs*) of a class called *Timer*.

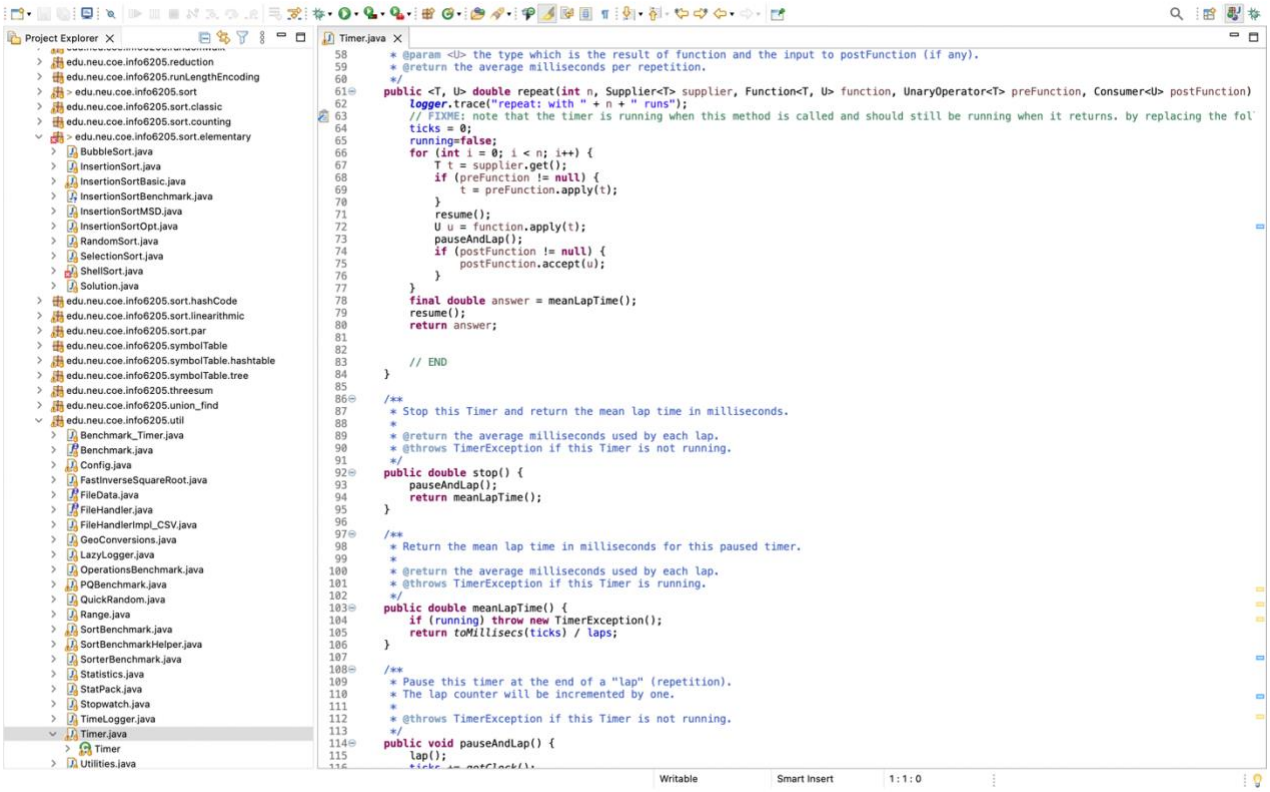
1)TIMER TEST CASES:



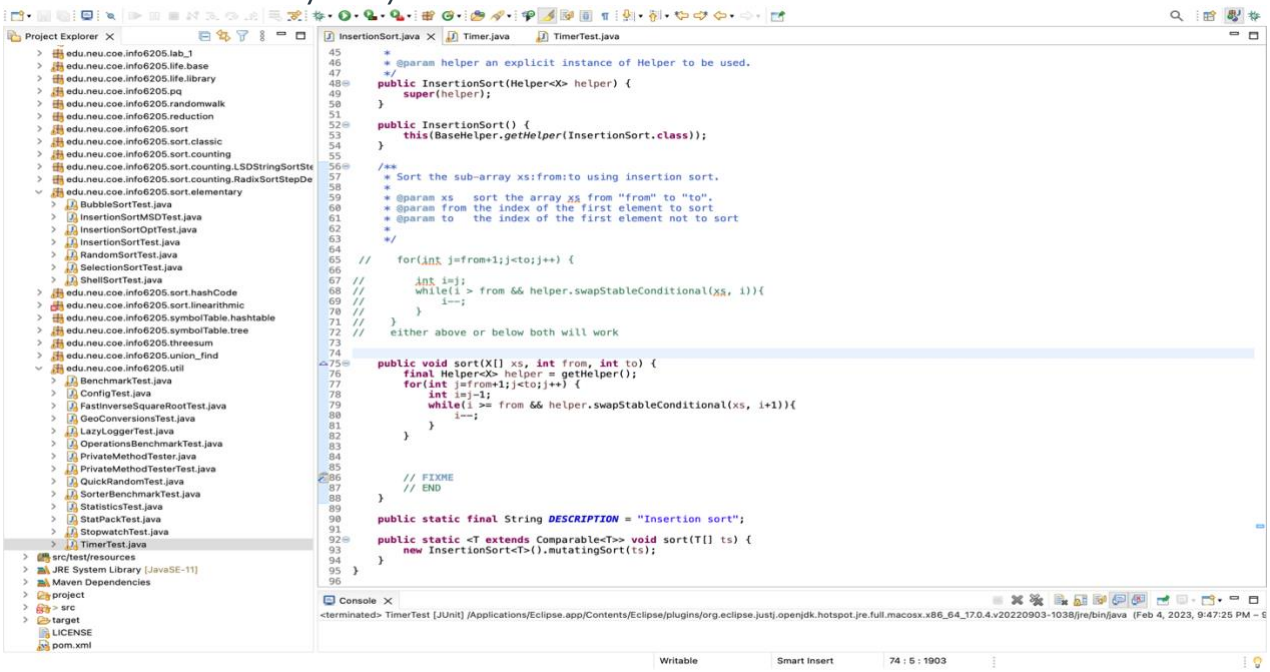
2)BENCHMARK TEST CASES:

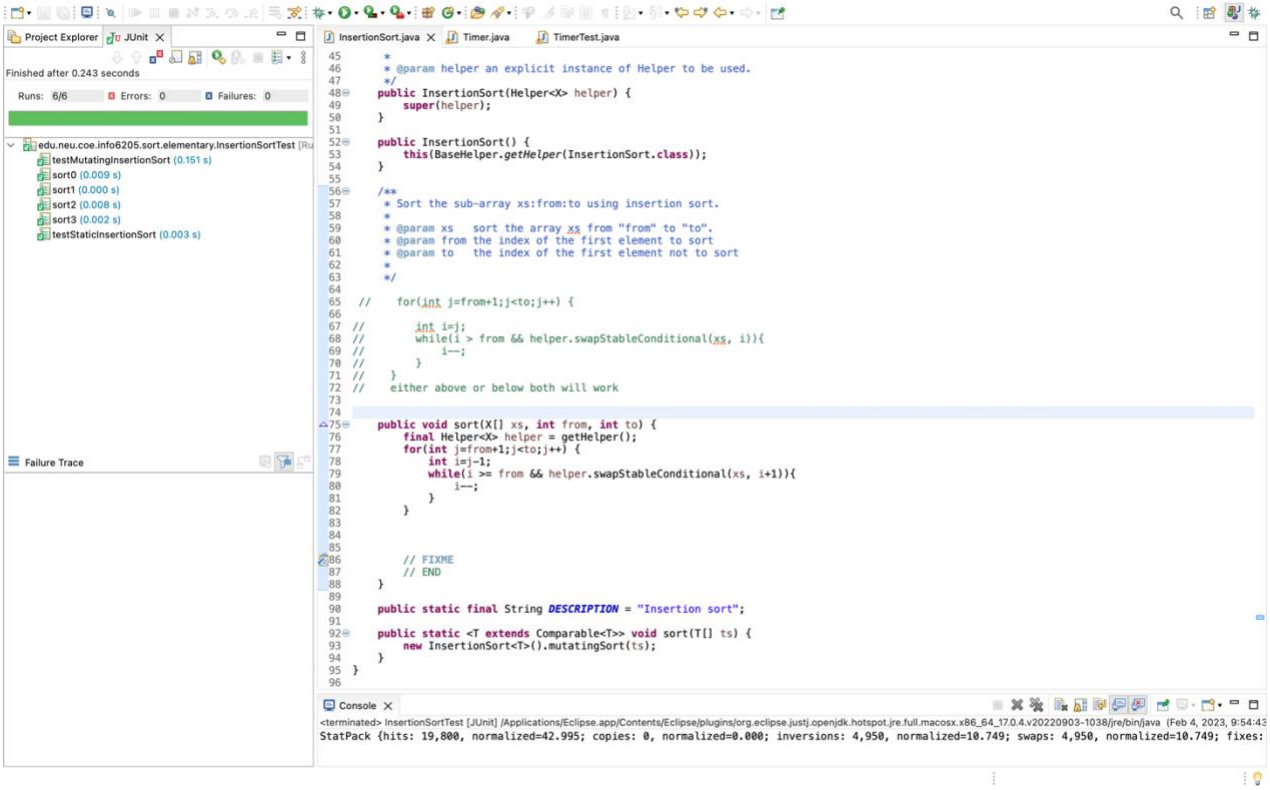


CODE TIMER:

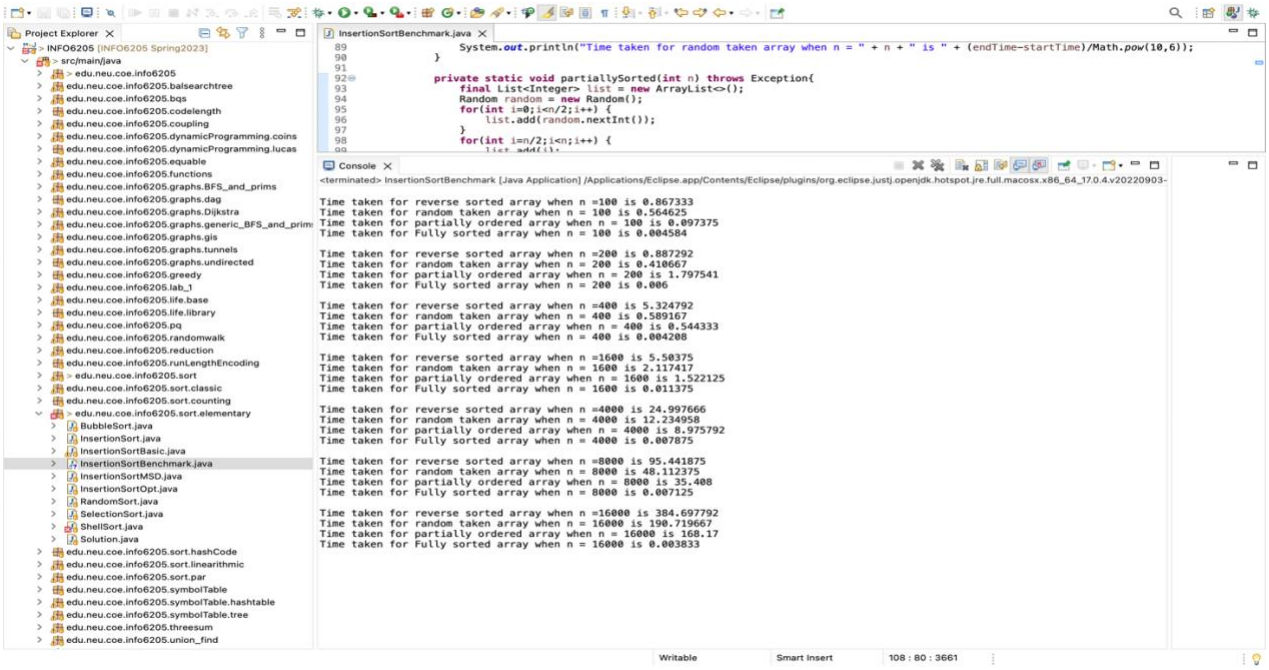


Task 2: Implementing *InsertionSort* (in the *InsertionSort* class) by simply looking up the insertion code used by *Arrays.sort*.





Task 3: Implementing the main program (or you could do it via your own unit tests) to actually measure the time for sorted , partially sorder, random , reverse arrays.



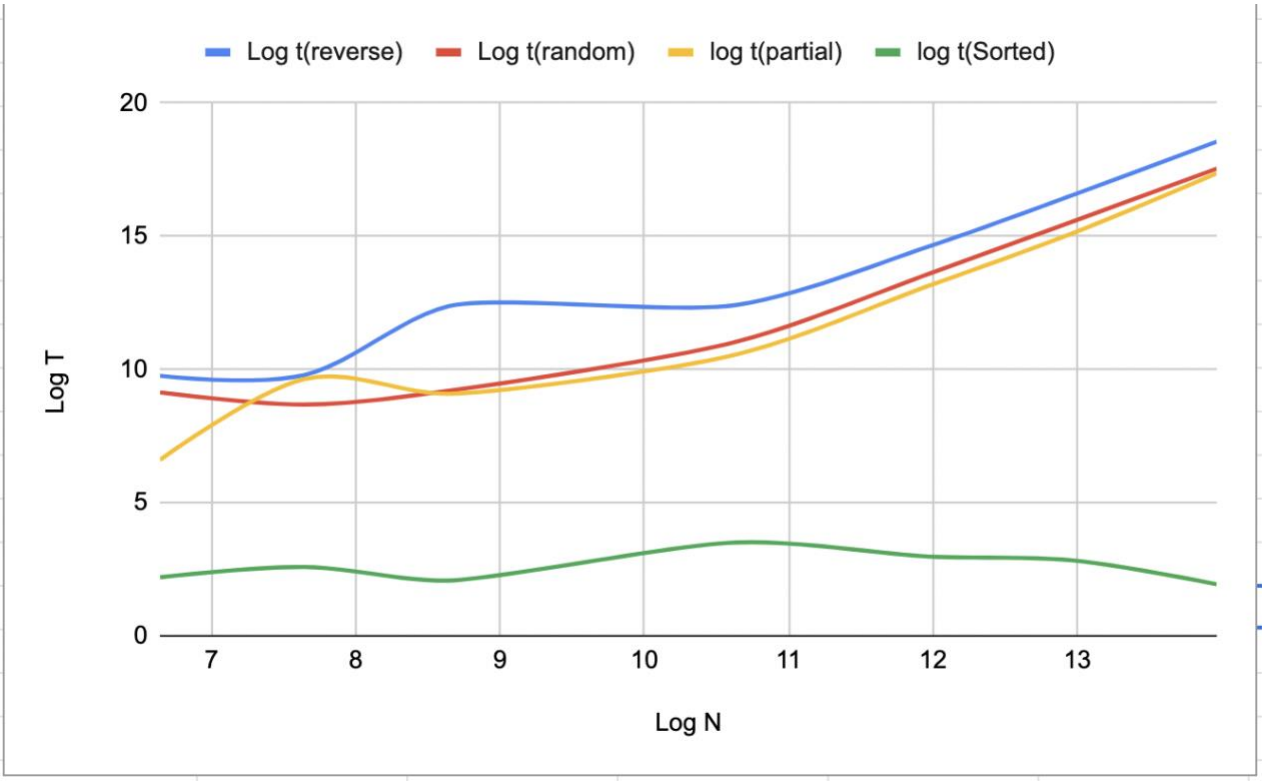
VALUES FOR N AND T:

	A	B	C	D	E	F	G	H	I	J	K
1	N	Log N	t(reverse) milli sec	Log t(reverse)	t(random) milli sec	Log t(random)	t (partial) milli sec	log t(partial)	t(Sorted) msec	log t(Sorted)	
2	100	6.64385619	0.867333	9.760442192	0.564625	9.141149198	0.097375	6.605479518	0.004584	2.196607044	
3	200	7.64385619	0.887292	9.793265151	0.410667	8.681825211	0.797541	9.639414877	0.006	2.584962501	
4	400	8.64385619	5.324792	12.37850946	0.589167	9.202532815	0.544333	9.088345691	0.004208	2.073134705	
5	1600	10.64385619	5.50375	12.42619922	2.117417	11.0480897	1.522125	10.57187113	0.011375	3.50779464	
6	4000	11.96578428	24.997666	14.60950578	12.234958	13.57872153	8.975792	13.13182353	0.007875	2.977279923	
7	8000	12.96578428	95.441875	16.54233477	48.112375	15.5541204	35.408	15.11178774	0.007125	2.832890014	
8	16000	13.96578428	384.697792	18.55336602	190.719667	17.5410941	168.17	17.35956084	0.003833	1.938473998	

VALUES FOR LOG N AND LOG T:

	A	B	C	D	E
1	**Log N**	**Log t(reverse)**	**Log t(random)**	**log t(partial)**	**log t(Sorted)**
2	6.64385619	9.760442192	9.141149198	6.605479518	2.196607044
3	7.64385619	9.793265151	8.681825211	9.639414877	2.584962501
4	8.64385619	12.37850946	9.202532815	9.088345691	2.073134705
5	10.64385619	12.42619922	11.0480897	10.57187113	3.50779464
6	11.96578428	14.60950578	13.57872153	13.13182353	2.977279923
7	12.96578428	16.54233477	15.5541204	15.11178774	2.832890014
8	13.96578428	18.55336602	17.5410941	17.35956084	1.938473998

Graph (log N vs Log t):



Conclusion:

The time taken for the array which is in reverse order will be more as it needs more looping to sort the array fully as it is entirely in reverse order. And the time taken between partially sorted and random array will be almost same, to be precise random array will take little more time compared to partially sorted as the partial sorted array will need a smaller number of loops (already sorted half of its part) to sort it fully compared to random sorted which will need more time(loops). And finally, the sorted array which is already sorted will take very less time as it is already sorted.