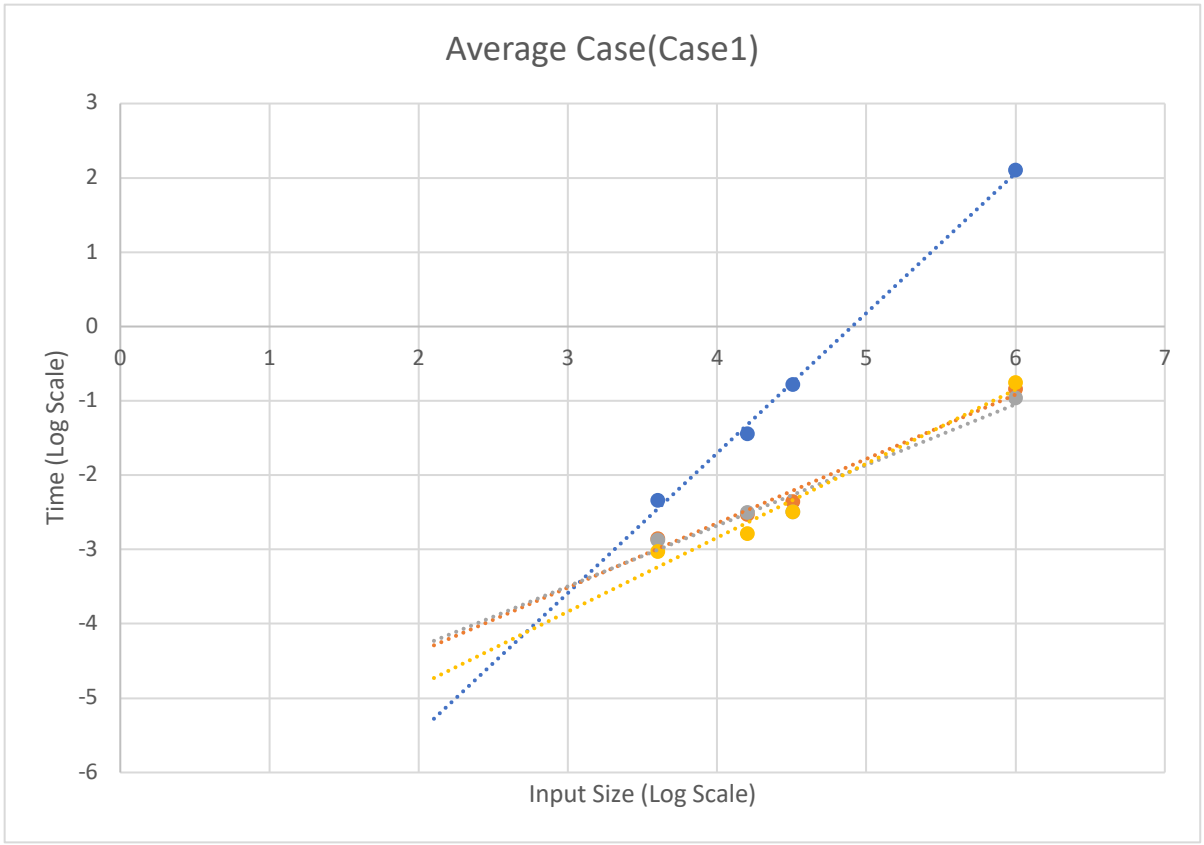
The program is written and ran in the EDA workstation.

Input Size	IS		MS		QS(Random)		HS	
	CPU time(s)	Memory(KB)	CPU time(s)	Memory(KB)	CPU time(s)	Memory(KB)	CPU time(s)	Memory(KB)
4000.case2	0.000105	5904	0.001039	5904	0.000906	5912	0.000537	5904
4000.case3	0.009747	5904	0.001025	5904	0.000897	5912	0.000802	5904
4000.case1	0.004577	5904	0.00141	5904	0.001347	5912	0.000936	5904
16000.case2	0.000172	6056	0.002077	6056	0.001244	6064	0.001933	6056
16000.case3	0.069115	6056	0.001196	6056	0.001494	6064	0.002062	6056
16000.case1	0.035897	6056	0.002939	6056	0.003166	6064	0.001647	6056
32000.case2	0.000162	6188	0.002843	6188	0.001756	6196	0.0026	6188
32000.case3	0.264808	6188	0.002391	6188	0.003124	6196	0.002344	6188
32000.case1	0.166983	6188	0.004371	6188	0.003196	6196	0.003216	6188
1000000.case2	0.001402	12144	0.070418	14004	0.06278	12152	0.077252	12144
1000000.case3	269.273	12144	0.066514	14004	0.065382	12152	0.09401	12144
1000000.case1	127.922	12144	0.143422	14004	0.108937	12152	0.176384	12144

Insertion sort: blue merge sort: orange quick sort(random): gray heap sort: yellow




Best Case(Case2)
Time (Log Scale)
Input Size (Log Scale)
0
-1
-2
-3
-4
-5
-6
0
1
2
3
4
5
6
7


Worst Case(Case3)
Time (Log Scale)
Input Size (Log Scale)
3
2
1
0
-1
-2
-3
-4
-5
-6
0
1
2
3
4
5
6
7

The plots are identical to the example given. The sorting time of insertion sort varies greatly from worst case to best case, while the time for the other sorting algorithms is invariant under different test cases. This is because insertion sort has best case time complexity $\Omega(n)$ and worst case complexity $O(n^2)$, while the other algorithms has time complexity $\Theta(n \log n)$. Therefore, the worst case time complexity for insertion sort has a higher slope than the other algorithms, which are steeper than the best case insertion sort. Looking at the slope, the worst case insertion sort has slope approximately 2, while the other algorithms and best case insertion sort has slope closer to 1.