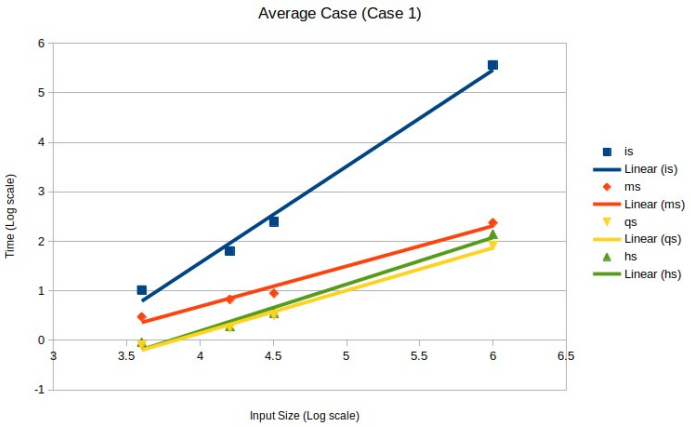
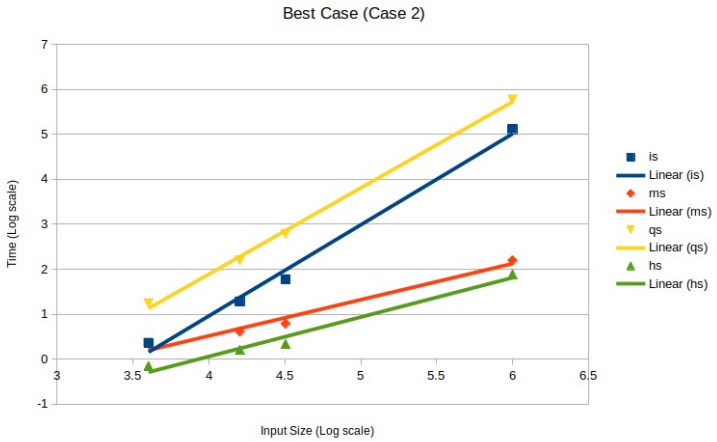
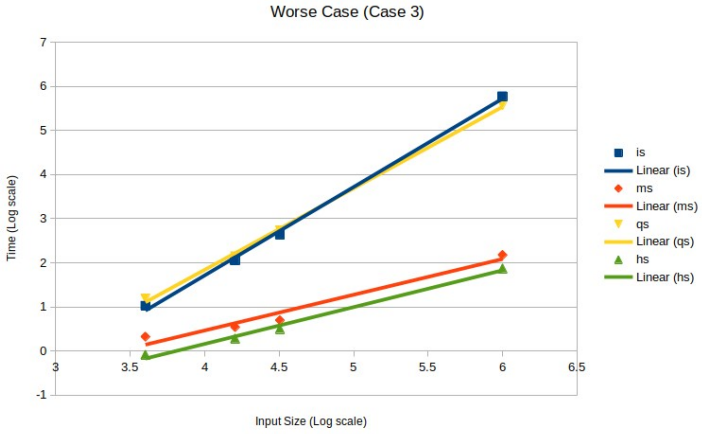
B09901186
Name: 賴群貿
Port: 40056

Input size	IS		MS		QS		HS	
	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)
4000.case2	2.283	5904	2.167	6040	17.352	5968	0.695	5904
4000.case3	10.512	5904	2.098	6040	15.283	5904	0.82	5904
4000.case1	10.319	5904	2.971	6040	0.8	5904	0.913	5904
16000.case2	19.023	6056	4.057	6056	155.968	6680	1.589	6056
16000.case3	113.055	6056	3.464	6056	139.176	6304	1.907	6056
16000.case1	63.44	6056	6.695	6056	1.791	6056	1.898	6056
32000.case2	59.15	6188	6.093	6316	608.724	7496	2.147	6188
32000.case3	433.87	6188	4.946	6316	541.945	6736	3.164	6188
32000.case1	246.241	6188	8.867	6316	3.22	6188	3.585	6188
1000000.case2	130144	12144	155.724	16236	590975	56840	76.294	12144
1000000.case3	590855	12144	149.352	16236	370012	27252	74.153	12144
1000000.case1	362288	12144	235.357	16236	81.955	12144	139.584	12144







1. The worse case(1000000) of QuickSort and InsertionSort takes a lot of time, and their slope is approximately 2 times higher than others', since their run time is $O(n^2)$.
2. Moreover, I was wondering that this experiment case classification is not quite right, since the same case can be the best case for some algorithms while others' don't.
3. The real best case for InsertionSort is really fast (0.116ms for 1000000 case), since it's best case run time is O(n)
4. MergeSort's and HeapSort's performances are quite good(smooth slope), but MergeSort will take more Memory.